

|     |       |      |                |                           |  |
|-----|-------|------|----------------|---------------------------|--|
| JRL | Vol.9 | No.2 | Hal. 113 - 126 | Jakarta,<br>Desember 2016 | ISSN : 2085.3866<br>No.376/AU1/P2MBI/07/2011 |
|-----|-------|------|----------------|---------------------------|--|

## **PERANCANGAN *DATABASE* HASIL ANALISA SWAPANTAU INSTALASI PENGOLAHAN AIR LIMBAH PADA INDUSTRI KEMASAN KALENG**

**Oman Sulaeman dan Imam Setiadi**  
Badan Pengkajian dan Penerapan Teknologi  
Email : oman.sulaeman@bppt.go.id

### **Abstrak**

Dalam melakukan pemantauan kualitas lingkungan disamping memerlukan data analisa juga penyajian data yang akurat sehingga mampu menampilkan suatu informasi yang berguna. Jika data-data tersebut tidak segera didapatkan maka dikhawatirkan perubahan kualitas yang semakin memburuk sehingga akan mengakibatkan kesulitan dalam pemulihannya. Perancangan Database Hasil Analisa Swapantau Instalasi Pengolahan Air Limbah pada industri pembuatan kemasan kaleng ini bertujuan untuk membuat suatu prosedur baru dalam hal penyimpanan dan pengelolaan data menggantikan metoda lama yang masih konvensional. Perancangan database ini juga dilengkapi graphic user interface ( GUI ) sebagai media interaksi pengguna dengan software database tersebut. Database ini dirancang dengan metode Logical database design dimana proses pembuatan suatu model informasi yang digunakan pada perusahaan berdasarkan pada model data yang spesifik, tetapi tidak tergantung dari Database Management System (DBMS). DBMS adalah software yang memungkinkan pemakai untuk mendefinisi, membuat, memelihara, dan mengontrol akses ke database. Hasil yang ingin dicapai dengan adanya database ini adalah pengguna dapat dengan mudah melakukan penyimpanan data, melakukan update data, mempercepat penelusuran data terdahulu, dan membantu dalam penyajian laporan. Dengan adanya database hasil analisa swapantau ini maka data dapat di simpan dengan baik dan aman pada database untuk di kelola dikemudian hari sebagai sumber data juga sebagai pelengkap dalam pelaporan swapantau instalasi pengolahan air limbah.

**kata kunci** : swapantau, database, DBMS, update data, IPAL

# SELF-MONITORING ANALYSIS RESULTS DATABASE DESIGN FOR WASTEWATER TREATMENT INSTALATION IN MANUFACTURING INDUSTRY PACKAGING CANS

## **Abstract**

*In monitoring the environmental quality, besides it needs the data analysis also requires the presentation of accurate data so as to display an useful information. If the data is not immediately available it is feared worsening quality changes that will lead to difficulties in recovery. Self-monitoring analysis results database design for wastewater treatment instalation in manufacturing industry packaging cans aims to create a new procedure in terms of the storage and management to create a new procedure in terms of the storage and management of data replaces the old method is still conventional. Database design also features graphic user interface (GUI) as a medium for user interaction with the database software. Database database software is designed with the Logical database design method in which the process of making a model of the information used in the company based on the model-specific data, but does not depend on the Database Management System (DBMS). DBMS is a software that allows users to define, create, maintain, and control access to the database. Results to be achieved with the existence of this database is that users can easily perform data storage, update the data, speed up retrieval of earlier data, and assist to present a report. Given this self-monitoring analysis results database, the data can be stored properly and safely on the database to be managed in the future as well as complementary data sources in reporting self-monitoring of wastewater treatment.*

**keywords :** *self monitoring, database,DMBS, data update, and wastewater treatment plant*

## I. PENDAHULUAN

Dalam pengolahan data secara manual di temui kendala seperti ketidakakuratan data, kehilangan data yang dikarenakan catatan rusak, penulisan yang tidak terbaca dan lain-lain. Perancangan sistem *database* ini diharapkan dapat mengatasi kendala tersebut. Perancangan *database* ini bertujuan agar pengolahan data yang dilakukan memenuhi kewajiban sebagaimana tertuang dalam PP NO 82 Tahun 2001 Tentang Pengelolaan Kualitas Air Dan Pengendalian Pencemaran Air tentang kewajiban melakukan swapantau dan kewajiban untuk melaporkan hasil swapantau.

Pada saat ini PT. UCC telah melakukan swapantau kualitas hasil olahan limbahnya. Saat ini pengumpulan data hasil analisa yang dilakukan masih menggunakan cara manual dengan menggunakan kertas sebagai media pencatatan dan buku laporan sebagai media penyimpanan datanya. Cara tersebut di samping tidak efisien pada waktu penyimpanan juga melibatkan banyaknya tumpukan – tumpukan kertas hasil analisa swapantau, sehingga masih menimbulkan masalah sebagai berikut:

1. Data tidak tersimpan dengan baik, kemungkinan kertas terpercik ketika sedang melakukan analisa sampel, sehingga resiko kehilangan data menjadi besar,
2. Tidak mudah melakukan penelusuran rekam data terdahulu,
3. Proses pembuatan laporan yang rumit sehingga membutuhkan waktu yang lama
4. Data cenderung hilangan dikarenakan penumpukan kertas hasil analisa.

Prosedur konvensional belum dapat memberikan solusi atas permasalahan yang ada. Dalam rangka pemenuhan kebutuhan penyimpanan data secara baik dan terstruktur sehingga dapat mencapai efisiensi kerja pada bagian instalasi pengolahan air limbah dan unit daur ulang PT. UCC maka perlu adanya suatu prosedur dengan pendekatan yang berbasis aplikasi dipandang relevan untuk menjawab permasalahan tersebut.

Berdasarkan identifikasi masalah di atas maka: dapat dirumuskan bagaimana menerapkan prosedur baru untuk menggantikan prosedur lama yang masih bersifat konvensional dalam penyimpanan data dan pembuatan laporan swapantau pada bagian Instalasi Pengolahan Air Limbah di PT. UCC.

Tujuan perancangan database hasil analisa swapantau Instalasi Pengolahan Air

Limbah ini adalah menyediakan fasilitas penyimpanan data hasil analisa swapantau, menyediakan data pendukung dalam penyajian laporan dan menampilkan informasi data hasil analisa swapantau.

## II. METODE PENELITIAN

Dalam melakukan perancangan desain database menggunakan 2 (dua) metode penelitian yaitu :

1. Teknik Pengambilan Data  
Pengumpulan data yang dilakukan yaitu dengan melakukan observasi , wawancara & studi pustaka.
2. Teknik Pengembangan *Software*  
Terdiri dari analisa kebutuhan *software*, analisa sistem berjalan & desain.

### 3.1. Teknik Pengambilan Data

Sebelum dapat melakukan perancangan diperlukan sebuah data yang dibutuhkan dalam pembangunan sebuah pekerjaan. Adapun pengumpulan data yang dilakukan yaitu melakukan, observasi & wawancara pada pengguna, dalam hal ini adalah bagian Instalasi Pengolahan Air Limbah PT. UCC. Observasi dilakukan dengan pengamatan secara langsung mengenai cara-cara pengumpulan data hasil analisa swapantau dan pembuatan laporan dan melakukan studi pustaka. Setelah tiga bulan data – data hasil analisa ditelusuri kembali dan diambil untuk dijadikan dokumen pelaporan per tigabulan (triwulan).

#### 2.2.1. Observasi

Observasi dilakukan dengan pengamatan secara langsung mengenai cara-cara pengumpulan data hasil analisa swapantau dan pembuatan laporan.

Pengambilan sampel dilakukan di beberapa titik pantau seperti pada *inlet* dan *outlet* Instalasi Pengolahan Air Limbah, serta *inlet* dan *outlet* unit daur ulang. Maksud dari pengambilan sampel ini adalah untuk dilakukan analisa kualitas air sebelum dan sesudah pengolahan agar baku mutu IPAL dapat tercapai dan dikendalikan sesuai dengan Keputusan Menteri Negara Lingkungan Hidup Nomor: KEP-51/MENLH/10/1995 Tentang Baku Mutu Limbah Cair Bagi Kegiatan Industri Pasal 6.

Pengambilan sampel dan analisa dilakukan dua kali dalam satu hari yang terbagi dalam dua *shift* yaitu parameter yang dianalisa meliputi debit air limbah dalam m<sup>3</sup>/hari, pH (derajat keasaman) ,TDS (total padatan

terlarut), TSS (total endapan terlarut), *turbidity* (kekeruhan), *hardness*, organik, *fluoride*, sulfida dan sulfat (Pergub DKI NO 122, 2005).

Ada 4 jenis sampel yang diambil yaitu 2 sampel *inlet outlet* Instalasi Pengolahan Air Limbah dan 2 sampel *inlet outlet* daur ulang. Dari tiap - tiap sampel ada 10 parameter yang dianalisa, maka didapatkan 40 data dari semua sampel dalam 1 *shift*. Pengambilan sampel dan analisa dilakukan 2 kali dalam 1 hari berarti ada 80 data terkumpul dalam 1 hari berarti 560 data terkumpul dalam tiap minggunya.

Pekerjaan tersebut dilakukan oleh operator unit Instalasi Pengolahan Air Limbah, setelah melakukan analisa maka data dari hasil analisa tersebut dicatat dan disimpan ke dalam buku laporan. Setelah melakukan pencatatan selama tiga bulan maka data tersebut diambil kembali untuk dilaporkan sebagai laporan triwulan.

### 2.2.2. Wawancara

Wawancara dilakukan dengan mengajukan beberapa pertanyaan yang berkaitan dengan aktivitas operator – operator Instalasi Pengolahan Air Limbah di dalam pengumpulan data. Sedangkan tugas pembuatan laporan dilakukan oleh koordinator Instalasi Pengolahan Air Limbah.

### 2.2.3. Studi Pustaka

Studi pustaka dilakukan dengan pengumpulan data dari kegiatan bisnis, hasil analisa swapantau dan perancangan basis data. Semua kegiatan bisnis yang dikomputerisasikan pasti membutuhkan sistem *database* sebagai media penyimpanan. *Database* tersebut selain digunakan untuk menyimpan data transaksi juga akan digunakan untuk menampilkan laporan yang bisa digunakan pihak manajemen sebagai sumber untuk pengambilan keputusan. (<http://ciku.typepad.com/blog/2009/12/database.html/-/07-05-2012/>).

Data hasil analisa swapantau merupakan informasi yang penting yang menjadi tolok ukur efektifitas kinerja instalasi pengolah air limbah. Penyimpanan data hasil analisa swapantau menjadi sangat penting untuk menghasilkan informasi yang dibutuhkan. Penerapan *database* hasil analisa swapantau instalasi pengolahan air limbah ini menggunakan pendekatan CBIS (*Computer Based Information System*) adalah sistem informasi yang menggunakan komputer untuk menangani pengolahan datanya. (Wahjono Heru Dwi, 2002)

Perancangan basis data merupakan

proses menciptakan perancangan untuk basis data yang akan mendukung operasi dan tujuan perusahaan (Connolly, 2002). Dalam merancang suatu basis data, digunakan metodologi-metodologi yang membantu dalam tahap perancangan basis data. Metodologi perancangan adalah pendekatan struktur dengan menggunakan prosedur, teknik, alat, serta bantuan dokumen untuk membantu dan memudahkan dalam proses perancangan. Dengan menggunakan teknik metode desain ini dapat membantu dalam merencanakan, mengatur, mengontrol, dan mengevaluasi *database development project* (Connolly, 2002).

## 3.2. Metode Pengembangan Sistem

Pengembangan *software* yang dilakukan dengan cara melakukan: analisa kebutuhan *software*, analisa sistem berjalan dan melakukan perancangan/desain *database*.

### 2.2.1. Analisa Kebutuhan Software

Analisa kebutuhan *software* dilakukan untuk menentukan *software* yang tepat sesuai dengan kemampuan *software* yang akan digunakan dalam merancang sistem informasi yang akan di buat.

*Software* yang digunakan dalam pembuatan sistem informasi ini adalah: a. *Visual basic* 6.0 digunakan untuk membuat desain tampilan aplikasi yang akan dibuat. B. MySQL digunakan untuk pengelolaan *database*. C. ADODB digunakan sebagai penghubung atau *conector MySql* dengan *visual basic*.

### 2.2.2. Analisa Sistem

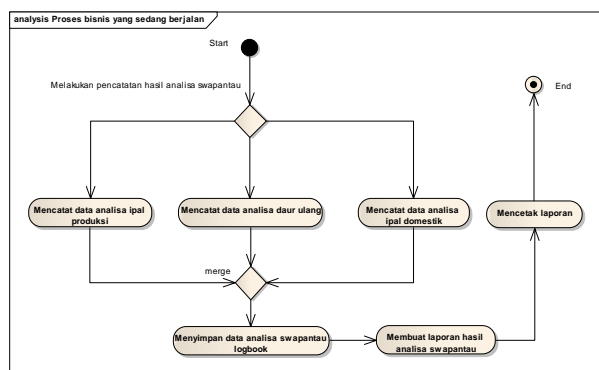
Dari uraian permasalahan di atas jelas tergambar bahwa prosedur penyimpanan data pada bagian instalasi pengolahan air limbah PT. UCC memerlukan waktu lama dan sulit melakukan penelusuran data kembali ketika data diperlukan dalam pembuatan laporan triwulan. Data cenderung tidak terawat, kertas rusak dan lain sebagainya.

Prosedur pengumpulan data hasil analisa swapantau adalah sebagai berikut:

1. Operator secara berkala pada saat yang ditentukan mengambil air sampel *inlet* dari unit Instalasi Pengolahan Air Limbah produksi melakukan analisa pada sampel tersebut.
2. Mencatat masing – masing hasil analisa dari 10 parameter yang ditetapkan pada selebar kertas, yang kemudian menjadi lembar data hasil analisa *inlet* Instalasi Pengolahan Air Limbah produksi.
3. Operator mengambil air sampel *outlet* Instalasi Pengolahan Air Limbah

4. Produksi, dan melakukan analisa.
5. Mencatat hasil analisa tersebut pada lembar catatan data hasil analisa *outlet* Instalasi Pengolahan Air Limbah produksi.
6. Operator melakukan perlakuan yang sama terhadap Instalasi Pengolahan Air Limbah Domestik dan unit daur ulang.
7. Setelah mendapat enam data hasil pengukuran *inlet* dan *outlet* dari masing – masing pengolah air, operator menyalin ke dalam sebuah buku laporan harian. Pada buku laporan harian juga tercatat kejadian – kejadian yang menyangkut kegiatan rutin bagian Instalasi Pengolahan Air Limbah, seperti pengadaan kimia, kerusakan *part* dari peralatan, dll.

Bagan alir proses bisnis yang sekarang berlangsung dapat dilihat pada Gambar 1.

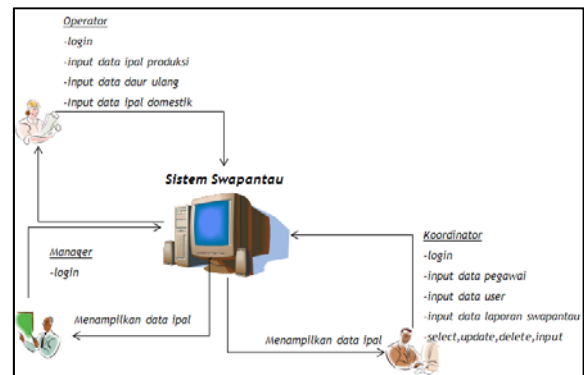


Gambar 1. Bagan Alir Proses yang Sedang Berlangsung

### 2.2.3. Perancangan Sistem

Perancangan *database* hasil analisa swapantau ini menggunakan pendekatan DBMS (*Database Managemen System*). *Database* hasil analisa swapantau ini secara khusus dibuat untuk memudahkan pemakai dalam mengelola *database* Instalasi Pengolahan Air Limbah. Sistem ini juga dibuat untuk mengatasi kelemahan sistem pemrosesan yang berbasis berkas.

Pada pendekatan yang berbasis berkas, umumnya perancangan sistem didasarkan kepada kebutuhan individual pemakai. Sehingga kemungkinan duplikasi data terdapat pada berkas – berkas tersebut. Konsep dari cara kerja *database* hasil analisa swapantau ini dapat dilihat pada Gambar 2.



Gambar 2. Diagram Kontek Database Hasil Analisa Swapantau

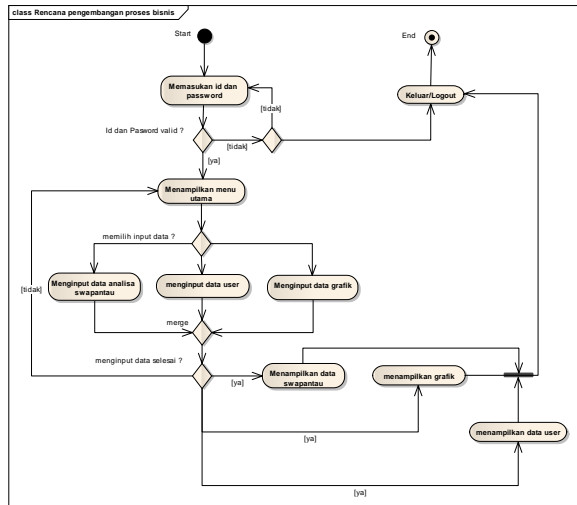
Deskripsi aktor merepresentasikan orang, peran atau sistem lain yang berinteraksi dengan sistem, aktor juga terhubung dengan *use case* melalui asosiasi. Untuk sistem *database* hasil analisa swapantau, aktor dapat diidentifikasi seperti pada Tabel 1 di bawah ini.

Tabel 1. Identifikasi Aktor

| Aktor               | Keterangan   |
|---------------------|--|
| Koordinator / Admin | Seorang yang mempunyai hak akses penuh untuk mengelola database seperti melihat, <i>input</i> , ubah, menampilkan grafik dan penambahan user |
| Operator            | Seorang yang mempunyai hak akses untuk <i>Login</i> ke menu <i>input</i> data dan menampilkan data   |
| Manager             | Seorang yang mempunyai hak akses untuk melihat data dan menampilkan grafik   |

Dengan prosedur yang baru, penyimpanan data analisa swapantau yang baru dapat dilihat pada Gambar 3, dengan menggunakan alat bantu analisis berupa bagan alir proses bisnis. Bagan alir proses bisnis yang digunakan untuk menggambarkan prosedur yang baru.

Bagan alir proses di bawah ini menjelaskan aktifitas *user* dari proses bisnis yang direncanakan mulai dari *login*, validasi dan verifikasi *username* dan *password*, menampilkan menu utama, memilih submenu, dan melakukan pengolahan data.



Gambar 3. Bagan Alir Proses yang Direncanakan

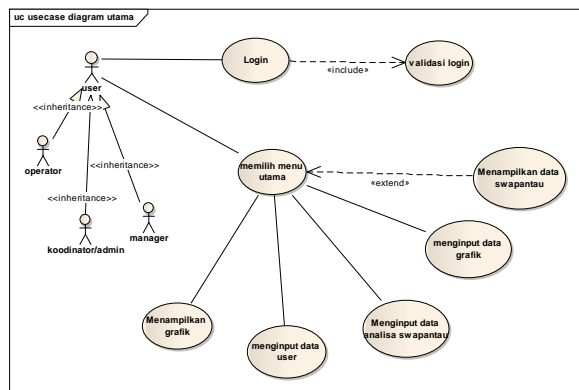
#### 2.2.4. Code Generation

*Code Generator* adalah “sebuah program yang dibuat agar dapat menghasilkan instruksi atau kode program sesuai dengan kebutuhan pengguna dalam ruang lingkup tertentu”. Pengguna cukup melakukan pengaturan seperlunya dan menekan tombol, maka lahirlah sebuah program lengkap dan berfungsi sesuai dengan konfigurasi pengguna, tanpa harus melewati tahap mencari dan memperbaiki *Error*. (terlampir)

### III. HASIL DAN PEMBAHASAN

#### 3.1. Usecase

*Usecase* digunakan sebagai perangkat untuk mendeskripsikan sistem, lingkungan dan relasi antara sistem dengan lingkungannya.



Gambar 4. Usecase diagram

Tabel 2. Deskripsi *Usecase Input Data*

|                             |   |
|-----------------------------|---|
| <i>Use Case name</i>        | Checkout  |
| <i>Requirement</i>          | A3,A4,A5  |
| <i>Pre-condition</i>        | User telah login  |
| <i>Post-condition</i>       | Data tersimpan,ter-update, atau terhapus  |
| <i>Failed end condition</i> | Gagal menyimpan, meng-update, atau terhapus   |
| <i>Main flow/basic path</i> | <ol style="list-style-type: none"> <li>1. user melihat form <i>input data</i>.</li> <li>2. user memilih tombol “tambah data”.</li> <li>3. sistem menampilkan <i>from data</i>,</li> <li>4. user menginput data baru.</li> <li>5. user memilih tombol “simpan”.</li> <li>6. sistem menyimpan data.</li> <li>7. sistem menutup <i>from data input</i>.</li> </ol> |

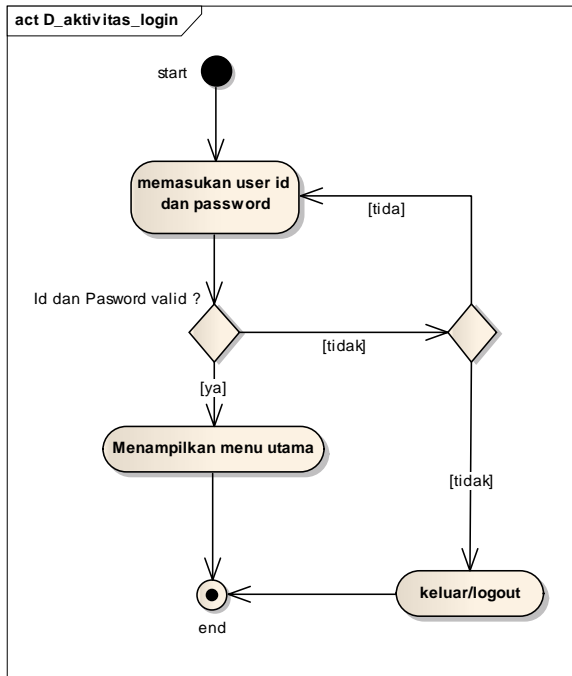
#### 3.2. Activity diagram

*Activity diagram* adalah sesuatu yang menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. Terdiri dari :

##### a. Proses login

Prosedur *login* diperlukan untuk membagi hak akses, apabila *user* adalah operator maka *user* memasukan *user id* dan *password* untuk operator. Operator bisa mendaftarkan *username* dan *passwordnya* oleh sistem administrator. Untuk lebih jelas lihat proses bisnis diagram aktifitas pada Gambar 5.

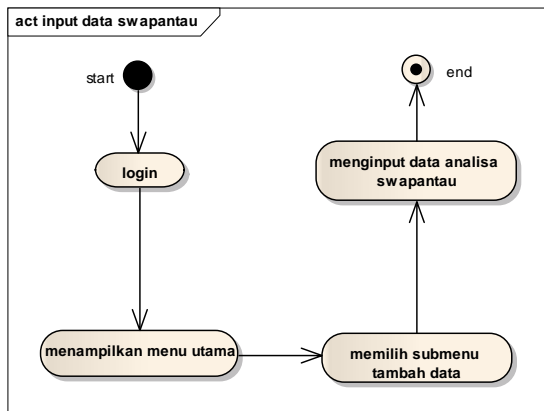




Gambar 5. Activity Diagram Proses Login

#### b. Input data :

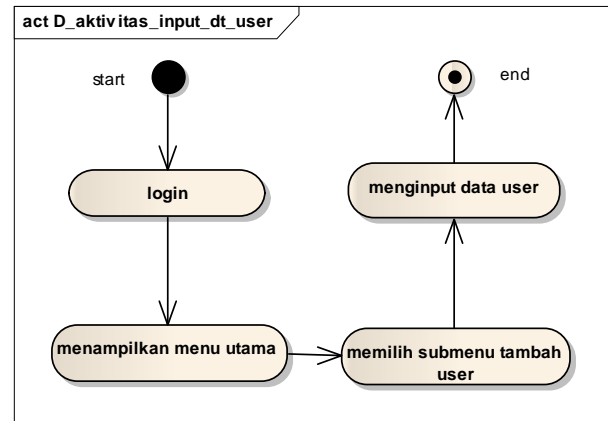
Bagian ini menjelaskan aktifitas *user* melakukan input data seperti dijelaskan pada Gambar 6. Proses ini berlaku berulang untuk masing – masing akun.



Gambar 6. Activity Diagram Input Data

#### c. Input Data User

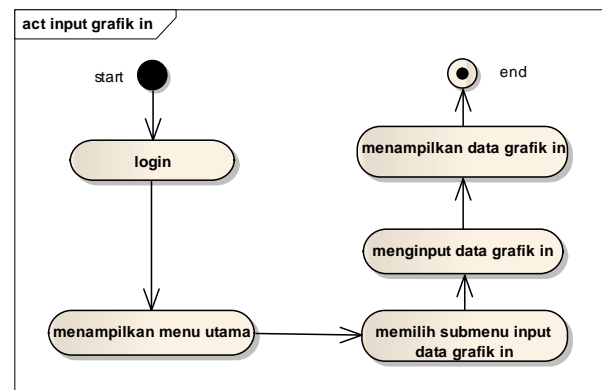
Bagian ini menjelaskan aktifitas *user* melakukan penambahan data user seperti dijelaskan pada Gambar 7. Proses hanya bisa dilakukan *user* dengan akun admin.



Gambar 7. Activity Diagram Tambah User

#### d. Input Data Grafik Inlet

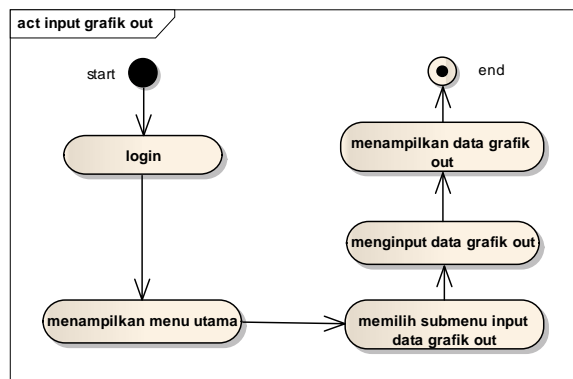
Bagian ini menjelaskan aktifitas *user* melakukan pemasukan data parameter *inlet* IPAL dan menampilkan grafik dijelaskan pada Gambar 8. Proses hanya bisa dilakukan *user* dengan akun admin dan *manager*.



Gambar 8. Activity Diagram Input Data Grafik Inlet

#### e. Input Data Grafik Outlet

Bagian ini menjelaskan aktifitas *user* melakukan pemasukan data parameter *outlet* IPAL dan menampilkan grafik dijelaskan pada Gambar 9. Proses hanya bisa dilakukan *user* dengan akun admin dan *manager*.



Gambar 9. Diagram Aktifitas *Input Data Grafik Outlet*

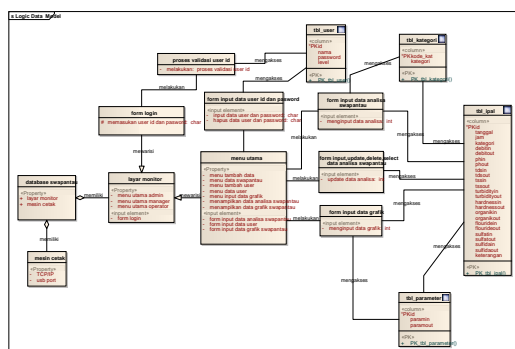
### 3.3. Desain

Dalam merancang *database* digunakan, 2 model data, yaitu:

- Model Logika Data (*Logical Data Model*), yaitu konsep bagaimana data dapat merepresentasikan sebuah kenyataan, dimasukkan ke dalam sebuah pemrosesan logika dan dapat menghasilkan informasi.
- Model Fisik Data (*Physical Data Model*), yaitu konsep bagaimana data disimpan pada media penyimpanan (*storage*) dalam suatu susunan secara fisik.

#### 3.3.1. Logical data model

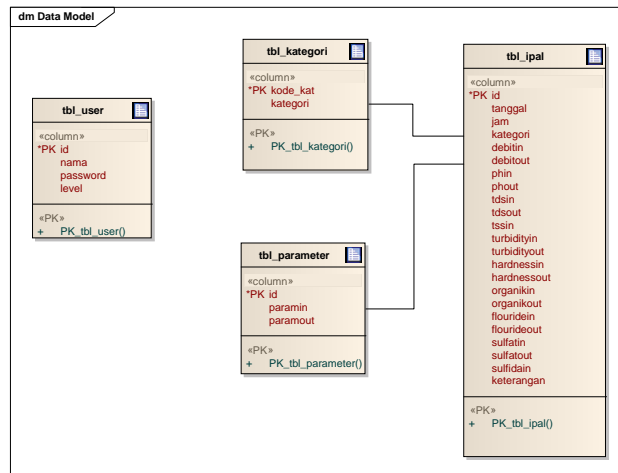
Menurut Connolly (2010, p467), perancangan basis data logikal adalah proses membangun model data yang digunakan di dalam suatu perusahaan, bersifat *independent* terhadap DBMS tertentu dan segala pertimbangan fisik lainnya. Tahap desain logikal *database* yang dimulai dengan membuat *maps konseptual database design* pada model data logikal. Model data logikal merupakan sumber informasi untuk tahapan desain fisik, seperti terlihat pada Gambar 10.



Gamba 10. Logical Data Model

#### 3.3.2. Physical data model

Menurut Connolly (2010, p467) perancangan basis data fisik adalah proses untuk menghasilkan deskripsi implementasi *database* pada penyimpanan sekunder (*secondary storage*) yang menggambarkan hubungan dasar, organisasi *file*, dan indeks yang digunakan untuk mendapatkan akses yang cepat terhadap data dan setiap integrity constraints terkait dan langkah-langkah keamanan yang ada.



Gambar 11. Physical Data Model

### 3.4. Software Architecture

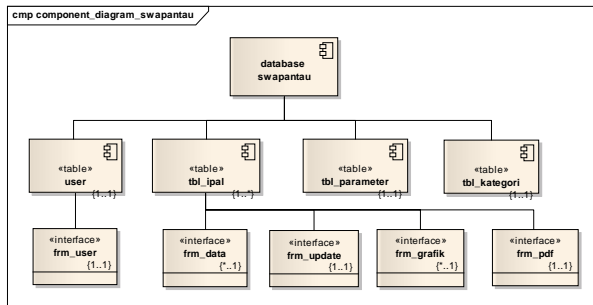
Arsitektur perangkat lunak adalah struktur sistem dari suatu program atau sistem komputer yang terdiri dari komponen-komponen perangkat lunak, ciri yang tampak secara eksternal dari komponen-komponen tersebut, serta hubungan antar komponen tersebut. Istilah ini juga merujuk pada dokumentasi arsitektur perangkat lunak suatu sistem.

Dengan terstrukturanya program yang dibuat maka digunakan *component diagram* & *deployment diagram*.

#### 3.4.1. Component Diagram

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan diantaranya. *Component Diagram* juga dapat berupa *interface* yang berupa kumpulan layanan yang disediakan oleh komponen untuk komponen lainnya.

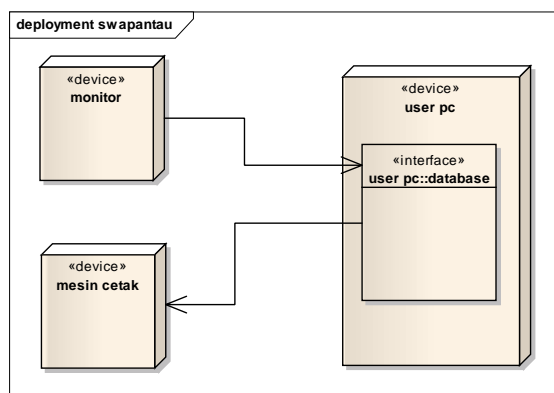




Gambar 12. Component Diagram Database Hasil Analisa Swapantau

### 3.4.2. Deployment Diagram

*Deployment Diagram* menggambarkan tata letak sistem secara fisik, yang menampilkan bagian-bagian *software* yang berjalan pada *hardware* yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen *hardware-hardware* tersebut.



Gambar 13. Deployment Diagram Database Hasil Analisa Swapantau Instalasi Pengolahan Air Limbah

## 3.5. Rancangan Layar Aplikasi

Rancangan Layar Aplikasi menggambarkan tampilan program terdiri dari: Tampilan *login*, tampilan halaman utama, tampilan *frame data*, tampilan *frame* lengkap & tampilan grafik *outlet* swapantau (Wahjono Heru Dwi, 2002).

### a. Tampilan Form Login

Setiap user yang terlibat akan diberikan *username* dan *password* yang berbeda-beda dengan hak akses yang berbeda pula. Jika user salah memasukkan *username* atau *password* maka sistem akan merespon dengan menampilkan suatu peringatan. Dan jika user benar memasukkan *username* dan *password* maka masuk pada menu utama dari masing-masing *account*. Seperti gambar 10 di bawah ini.

Gambar 14. Tampilan Form Login

### b. Tampilan Halaman Utama

Jika user akan melakukan proses *input* data maka user harus memilih sub menu "tambah data". Kemudian operator dapat melihat hasil *input* data barunya dengan mengklik submenu data swapantau maka sistem menampilkan *form input* data swapantau seperti Gambar 11 di bawah ini.

Gambar 15. Tampilan Halaman Utama

### c. Tampilan Frame Data

Selain itu bila operator ingin menampilkan data swapantau dengan tampilan penuh, operator dapat mengklik *link* detail, maka sistem akan menampilkan *frame data* lengkap, seperti Gambar 16 dan 17, di bawah ini.

| ORGANSK OUT | FLOURE IN | FLOURE OUT | SULFAT IN | SULFAT OUT | SULFIDA IN | SULFIDA OUT | KETERANGAN | DETAIL |
|-------------|-----------|------------|-----------|------------|------------|-------------|------------|--------|
| 5           | 89        | 4          | 89        | 48         | 4          | 8           | nambah     | Detail |
| 49          | 49        | 7          | 8         | 9          | 9          | 4           | ryan       | Detail |
| 8           | 9         | 4          | 9         | 5          | 9          | 8           | si mbah    | Detail |
| 2           | 9         | 1          | 9         | 4          | 9          | 0           | rya        | Detail |
| 12          | 9         | 3          | 9         | 4          | 9          | 4           | had        | Detail |
| 4           | 9         | 6          | 9         | 7          | 9          | 2           | oparti     | Detail |
| 6           | 9         | 4          | 9         | 1          | 9          | 2           | ary        | Detail |
| 16          | 9         | 7          | 9         | 8          | 9          | 7           | selah      | Detail |
| 4           | 9         | 39         | 9         | 2          | 9          | 1           | iman       | Detail |
| 6           | 9         | 2          | 9         | 7          | 9          | 4           | iman       | Detail |
| 7           | 9         | 4          | 9         | 3          | 9          | 5           | had        | Detail |
| 19          | 9         | 7          | 9         | 5          | 9          | 6           | iman       | Detail |
| 3           | 9         | 4          | 9         | 6          | 9          | 7           | hengeng    | Detail |
| 7           | 9         | 6          | 9         | 8          | 9          | 5           | mbah       | Detail |
| 2           | 9         | 5          | 9         | 8          | 9          | 6           | selah      | Detail |
| 6           | 9         | 1          | 9         | 8          | 9          | 4           | selah      | Detail |
| 6           | 9         | 4          | 9         | 8          | 9          | 5           | selah      | Detail |
| 2           | 9         | 8          | 9         | 4          | 9          | 4           | selah      | Detail |
| 8           | 9         | 2          | 9         | 1          | 9          | 6           | nodron     | Detail |
| 1           | 1         | 1          | 89        | 2          | 79         | 3           | nodron     | Detail |
| 7           | 9         | 3          | 9         | 1          | 9          | 3           | eri        | Detail |
| 6           | 9         | 8          | 9         | 4          | 9          | 3           | had        | Detail |

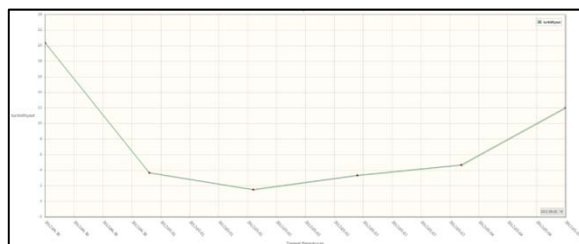
Gambar 16. Tampilan *Frame Data*

| NO | NAMA   | DAN    | KAWAN  | ... |
|----|--------|--------|--------|-----|
| 1  | 101101 | 101101 | 101101 | ... |
| 2  | 101101 | 101101 | 101101 | ... |
| 3  | 101101 | 101101 | 101101 | ... |
| 4  | 101101 | 101101 | 101101 | ... |
| 5  | 101101 | 101101 | 101101 | ... |
| 6  | 101101 | 101101 | 101101 | ... |
| 7  | 101101 | 101101 | 101101 | ... |
| 8  | 101101 | 101101 | 101101 | ... |
| 9  | 101101 | 101101 | 101101 | ... |
| 10 | 101101 | 101101 | 101101 | ... |
| 11 | 101101 | 101101 | 101101 | ... |
| 12 | 101101 | 101101 | 101101 | ... |
| 13 | 101101 | 101101 | 101101 | ... |
| 14 | 101101 | 101101 | 101101 | ... |
| 15 | 101101 | 101101 | 101101 | ... |
| 16 | 101101 | 101101 | 101101 | ... |
| 17 | 101101 | 101101 | 101101 | ... |
| 18 | 101101 | 101101 | 101101 | ... |
| 19 | 101101 | 101101 | 101101 | ... |
| 20 | 101101 | 101101 | 101101 | ... |

Gambar 17. Tampilan *Frame Data* Lengkap

#### d. Tampilan Grafik *Outlet Database*

Pada sesi ini operasi input data grafik dan menampilkan grafik dapat dilakukan juga oleh dua *user* yaitu Administrator dan Manager. Kedua – duanya mendapat hak dan otoritas yang sama dalam aktifitas ini. Seperti dalam gambar 18 di bawah ini.



Gambar 18. Tampilan Grafik *Outlet Database* Swapantau

## IV. KESIMPULAN

Setelah menguraikan pembahasan tentang perancangan *database* hasil analisa swapantau instalasi pengolahan air limbah, maka dapat mengambil beberapa dampak

positif dan negatif antara lain:

Dampak Positif

- Meningkatkan kecepatan dalam menginput data analisa swapantau
- Menurunkan biaya pengadaan kertas dan buku
- Mempercepat dalam pembuatan grafik laporan
- Memudahkan dalam proses penelusuran data terdahulu
- Didapatkannya data analisa digital pada *database* sehingga mudah dalam mobilisasi data.

Dampak Negatif

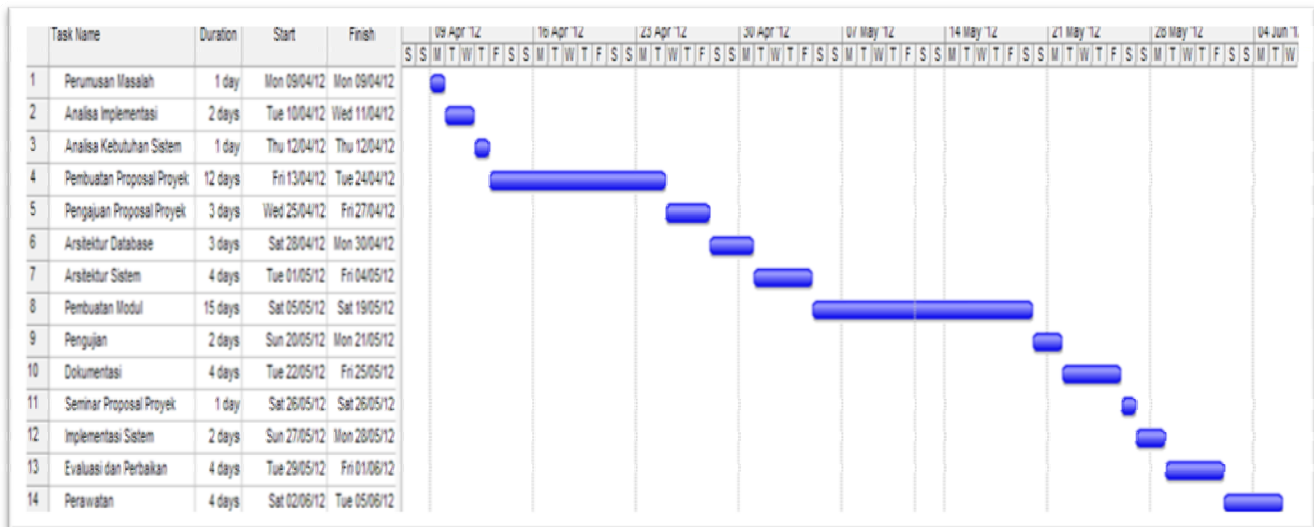
- Biaya perawatan sistem

Diharapkan program ini dapat digunakan dengan baik dan dikembangkan menjadi lebih baik dengan penambahan menu data yang lebih banyak.

## DAFTAR PUSTAKA

- Anonim (1). PP NO 82 Tahun 2001 Tentang Pengelolaan Kualitas Air Dan Pengendalian Pencemaran Air
- Anonim (2). KEP-51/MENLH/10/1995 Tentang Baku Mutu Limbah Cair Bagi Kegiatan Industri Pasal 6
- Wahjono Heru Dwi. 2002. Perancangan Sistem Basisdata Sumber Daya Air (Connolly.2002.p279). (Connolly.2002.p418). <http://ciku.typepad.com/blog/2009/12/database.html>

LAMPIRAN :



Gambar 19. Jadwal Pelaksanaan Pengambilan Sampel

Tabel 3. Usecase Narrative Login

TANGGAL : 25/04/2012 VERSI : 01.pb.swa1

|  |  |  |
|--|--|--|
| Nama Use Case :                        | Login  | Type Use Case  |
| ID Use Case :                          | 1.1  | Persyaratan Bisnis <input checked="" type="checkbox"/> |
| Prioritas :                            | Tinggi   |  |
| Sumber :                               | -  |  |
| Pelaku Bisnis Utama :                  | User   |  |
| Pelaku Partisipan Lain :               | Sistem   |  |
| Stakeholder yang berminat lain :       | -  |  |
| Deskripsi :                            | Mendeskripsikan bagaimana user berinteraksi dengan sistem, user melakukan login untuk masuk ke menu aplikasi sesuai dengan hak akses user.   |  |
| Prakondisi :                           | Untuk dapat masuk ke dalam sistem user terlebih dahulu mendaftarkan sebagai admin ataupun sebagai member.  |  |
| Sasaran :                              | Use Case ini diawali pada saat sistem dimulai yang bertujuan untuk proses validasi   |  |
| Bidang khas suatu event :              | -  | -  |
| Bidang alternatif :                    | -  | -  |
| Kesimpulan :                           | Use Case ini menyimpulkan bagaimana user masuk ke dalam menu sesuai dengan hak akses   |  |
| Pasca Kondisi :                        | User dapat masuk kedalam sistem apabila terdapat kecocokan pada saat validasi antara user Id dan password yang ada didalam database dengan user Id dan Password yang diinput user. |  |
| Aturan Bisnis :                        | Agar user dapat berinteraksi dengan sistem maka user terlebih dahulu harus melewati tahap ini  |  |
| Batasan dan spesifikasi implementasi : | Interface yang disediakan khusus untuk user yang telah terdaftar dan memiliki hak akses  |  |
| Asumsi :                               |  |  |
| Masalah Terbuka :                      | Bagaimana mengenal pentingnya keamanan suatu sistem.   |  |

## Lampiran sebagian Code Generatiao :

### CODE GENERATION

```
<?php
@session_start();
include 'konek.php';
if (isset($_POST['login']))
{
    $sql="select * from tbl_user where nama='".$_POST[username]."'";
    $query=mysql_query($sql);
    $rek=mysql_fetch_array($query);
    if (mysql_num_rows($query)>0 && $rek[password]==$_POST[password] &&
$rek[level]=='1' )
{header('location:/aer/module/opr/index.php');}
if (mysql_num_rows($query)>0 &&
$rek[password]==$_POST[password] && $rek[level]=='2' )
{header('location:/aer/module/koor/index.php');}
if (mysql_num_rows($query)>0 &&
$rek[password]==$_POST[password] && $rek[level]=='3' )
{header('location:/aer/module/mgr/index.php'); }
if (file_exists(username)); else
echo $validasi="<tr><h1>validasi login
gagal....!</h1></tr><tr><h1>Hubungi Admin Anda....!</h1></tr>";
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Database_Swapantau</title>
<!--
.style1 {color: #FFFFFF;
font-family: Geneva, Arial, Helvetica, sans-serif;
font-size: 12px;
font-weight: bold;}
.style3 {font-family: Geneva, Arial, Helvetica, sans-serif; font-size: 12px;}
-->
</style>
<style type="text/css">
<!--
.style2 {font-size: 12px;
color: #FFFFFF;
font-family: Geneva, Arial, Helvetica, sans-serif;
font-weight: bold;}
.style26 {font-family: Geneva, Arial, Helvetica, sans-serif; }
.style29 {font-family: Geneva, Arial, Helvetica, sans-serif; font-size: 11px; }
.style30 {font-size: 11px}
-->
</style>
</head>
<center>
<body>
<form id="form1" name="form1" method="post">
<table width="261" border="0">
<tr>
<td colspan="3" bgcolor="#40A0DB"><div align="center"
class="style2">MASUKAN USERNAME DAN PASSWORD ANDA</div></td>
</tr>
```

```

        <tr>
            <td width="64"><span class="style3 style26
style30">Username</span></td>
                <td width="9"><div align="center" class="style29">:</div></td>
            <td width="174"><input name="username" type="text" class="style29"
id="username" /></td>
                
            </tr>
            <tr>
                <td><span class="style3 style26 style30">Password</span></td>
                <td><div align="center" class="style29">:</div></td>
                <td><input name="password" type="password" class="style29"
id="password" /></td>
            </tr>
            <tr>
                <td><span class="style30"></span></td>
                <td><span class="style30"></span></td>
                <td><input name="login" type="submit" class="style29" id="login" value="Masuk"
/></td>
            </tr>
        </table>
    </form>
</body>
</html>

```

