



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kedudukan penulis selama kerja magang di PT. Kencana Cisadane Indonesia adalah sebagai *Junior Programmer* yang diberikan tugas langsung dari Manager Operasional PT. Kencana Cisadane Indonesia dan dikoordinasi langsung oleh Manager Operasional PT. Kencana Cisadane Indonesia.

#### 3.2 Tugas yang Dilakukan

Tugas yang dilakukan penulis selama kerja magang terfokus pada pembuatan perangkat lunak (*Software*) “*TASK MANAGEMENT SYSTEM*” yang akan digunakan untuk menunjang proses bisnis perusahaan. Rincian dari tugas-tugas yang dilakukan penulis adalah sebagai berikut :

Tabel 3.1 Rincian Tugas

Minggu Ke-	Jenis Pekerjaan yang Dilakukan
1	<ul style="list-style-type: none"><li>- Melakukan <i>requirement analysis</i> dan <i>functional specification</i> dari perangkat lunak yang akan dibuat.</li><li>- Melakukan presentasi sistem yang akan dibuat.</li><li>- Me-revisi <i>requirement dan functional specification</i>.</li></ul>
2	<ul style="list-style-type: none"><li>- Melakukan presentasi dari revisi yang telah dibuat.</li><li>- Membuat <i>Data Flow Diagram</i> (DFD).</li></ul>
3	<ul style="list-style-type: none"><li>- Membuat <i>Entity Relationship Diagram</i> (ERD).</li><li>- Melakukan implementasi model yang sudah dibuat ke dalam database (menggunakan <i>MySQL</i>).</li></ul>

Tabel 3.1 Rincian Tugas (lanjutan)

Minggu Ke-	Jenis Pekerjaan yang Dilakukan
4	<ul style="list-style-type: none"> <li>- Melakukan desain <i>interface</i> untuk modul <i>Login</i>.</li> <li>- Melakukan desain <i>interface</i> untuk modul <i>Master Data</i>.</li> <li>- Melakukan presentasi dari <i>interface-interface</i> yang telah dibuat.</li> <li>- Melakukan revisi dari <i>interface</i> pada modul <i>Master Data</i>.</li> </ul>
5	<ul style="list-style-type: none"> <li>- Melakukan <i>coding</i> untuk modul <i>Login</i>.</li> <li>- Melakukan <i>coding</i> untuk modul <i>Master Data</i>.</li> </ul>
6	<ul style="list-style-type: none"> <li>- Melanjutkan <i>coding</i> untuk modul <i>Master Data</i>.</li> <li>- Melakukan presentasi untuk modul <i>Master Data</i>.</li> <li>- Melakukan revisi pada modul <i>Master Data</i>.</li> </ul>
7	<ul style="list-style-type: none"> <li>- Melakukan desain <i>interface</i> untuk modul <i>Project</i></li> <li>- Melakukan presentasi dari <i>interface</i> pada modul <i>Project</i>.</li> <li>- Melakukan revisi dari <i>interface</i> pada modul <i>Project</i>.</li> </ul>
8	<ul style="list-style-type: none"> <li>- Melakukan <i>coding</i> untuk modul <i>Project</i>.</li> </ul>
9	<ul style="list-style-type: none"> <li>- Melanjutkan <i>coding</i> untuk modul <i>Project</i>.</li> </ul>
10	<ul style="list-style-type: none"> <li>- Melakukan presentasi dari modul <i>Project</i>.</li> <li>- Melakukan revisi pada modul <i>Project</i>.</li> </ul>
11	<ul style="list-style-type: none"> <li>- Melanjutkan revisi pada modul <i>Project</i>.</li> <li>- Melakukan integrasi dari modul-modul yang telah dibuat.</li> <li>- Melakukan <i>testing</i> menyeluruh pada perangkat lunak yang telah dibuat.</li> <li>- Melakukan perbaikan pada bug-bug yang ditemukan.</li> </ul>
12	<ul style="list-style-type: none"> <li>- Melakukan implementasi perangkat lunak yang telah dibuat ke unit-unit komputer perusahaan.</li> <li>- Melakukan <i>testing</i> perangkat lunak di setiap unit komputer perusahaan.</li> </ul>
13	<ul style="list-style-type: none"> <li>- Melakukan <i>training</i> untuk karyawan mengenai cara menggunakan perangkat lunak yang telah dibuat.</li> </ul>

### 3.3 Uraian Pelaksanaan Kerja magang

#### 3.3.1 Proses Pelaksanaan

Selama kurang lebih 3 bulan kerja magang di PT. Kencana Cisadane Indonesia, proses pelaksanaan kerja magang dalam membuat *Task Management System* secara umum terbagi menjadi 5 tahap. Tahap-tahap tersebut dapat dilihat melalui gambar berikut ini.



Gambar 3.1 Proses Kerja

Proses kerja magang dimulai dari tahap *requirement analysis* dan *functional specification* untuk mengetahui kebutuhan user, desain struktur sistem serta *interface*, *coding* untuk setiap fungsi yang dibutuhkan, *testing* setiap fungsi yang telah selesai dan kemudian implementasi pada komputer perusahaan. Untuk penjelasan yang lebih dalam mengenai tahap-tahap yang dilakukan dalam kerja magang akan dijelaskan dalam sub-bab berikut ini.

#### **A. Requirement Analysis dan Functional Specification**

Pada tahap *requirement analysis* dan *functional specification* penulis melakukan analisis atas apa yang dibutuhkan oleh perusahaan dan menentukan fungsi-fungsi yang terdapat di dalam sistem. Proses ini dilakukan mulai dari pemahaman akan proses bisnis dari perusahaan, mencari permasalahan dan kemudian mencari solusi atas permasalahan yang ada. PT. Kencana Cisadane Indonesia memiliki permasalahan dalam pendataan pelanggan, pendataan syarat untuk membuat suatu surat izin, pendataan proyek pembuatan surat izin, pendataan dan pengaturan tugas-tugas yang akan dilakukan dalam pembuatan surat izin. Setelah melakukan *requirement analysis* dan *functional specification* maka didapatkanlah beberapa modul yang akan dibuat di dalam sistem, yaitu modul *Master data* yang meliputi :

- Modul data pelanggan, yaitu modul yang dapat memanipulasi data pelanggan dan informasi mengenai pelanggan yang menggunakan jasa perusahaan.

- Modul data user, yaitu modul yang dapat memanipulasi data user dan informasi user yang dapat menggunakan sistem dengan hak-hak khusus.
- Modul data surat izin, yaitu modul yang dapat memanipulasi data surat izin yang akan dibuat oleh perusahaan sebagai bagian dari proyek.
- Modul data persyaratan surat izin, yaitu modul yang dapat memanipulasi data-data persyaratan dalam membuat sebuah surat izin.

Dan modul *Project* yang meliputi :

- Modul Proyek, yaitu modul yang digunakan untuk mengatur data proyek dan informasi proyek yang akan dibuat beserta tugas-tugas yang akan dibuat di dalam sebuah proyek.
- Modul Sisa Tugas, yaitu modul yang digunakan untuk menampilkan halaman yang menyajikan sisa-sisa tugas dari berbagai proyek yang masih belum terselesaikan.

## **B. Desain**

Pada tahap desain dimulai dengan melakukan desain struktur untuk perangkat lunak yang dimulai dengan membuat :

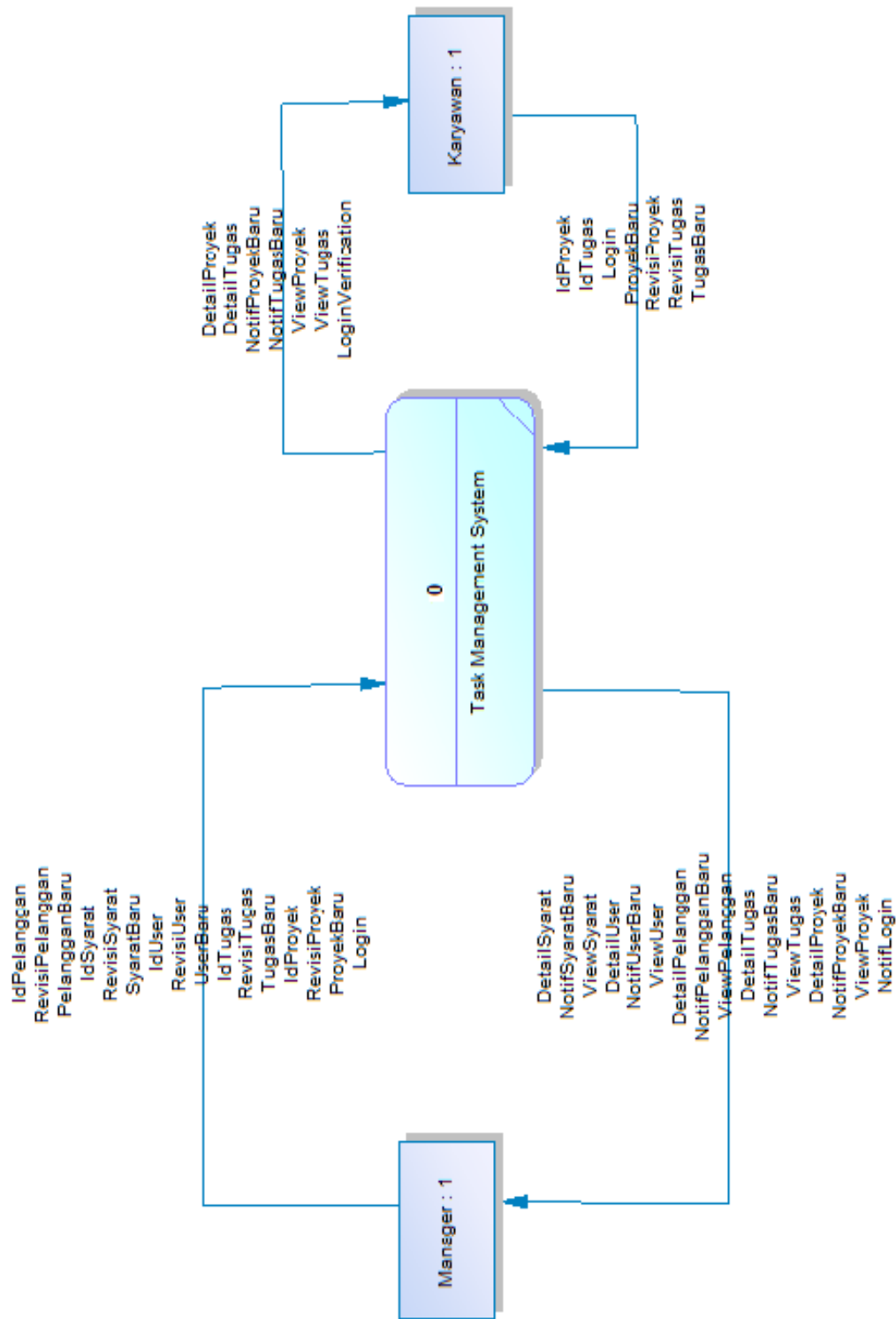
### **I. *Data Flow Diagram***

*Data Flow Diagram* (DFD), yaitu diagram yang menggambarkan suatu sistem yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik

dimana data itu mengalir (misalnya lewat telepon, surat dan sebagainya) dan disimpan (misalnya diskette, harddisk dan sebagainya). DFD menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas. Tiga kelebihan utama DFD yaitu :

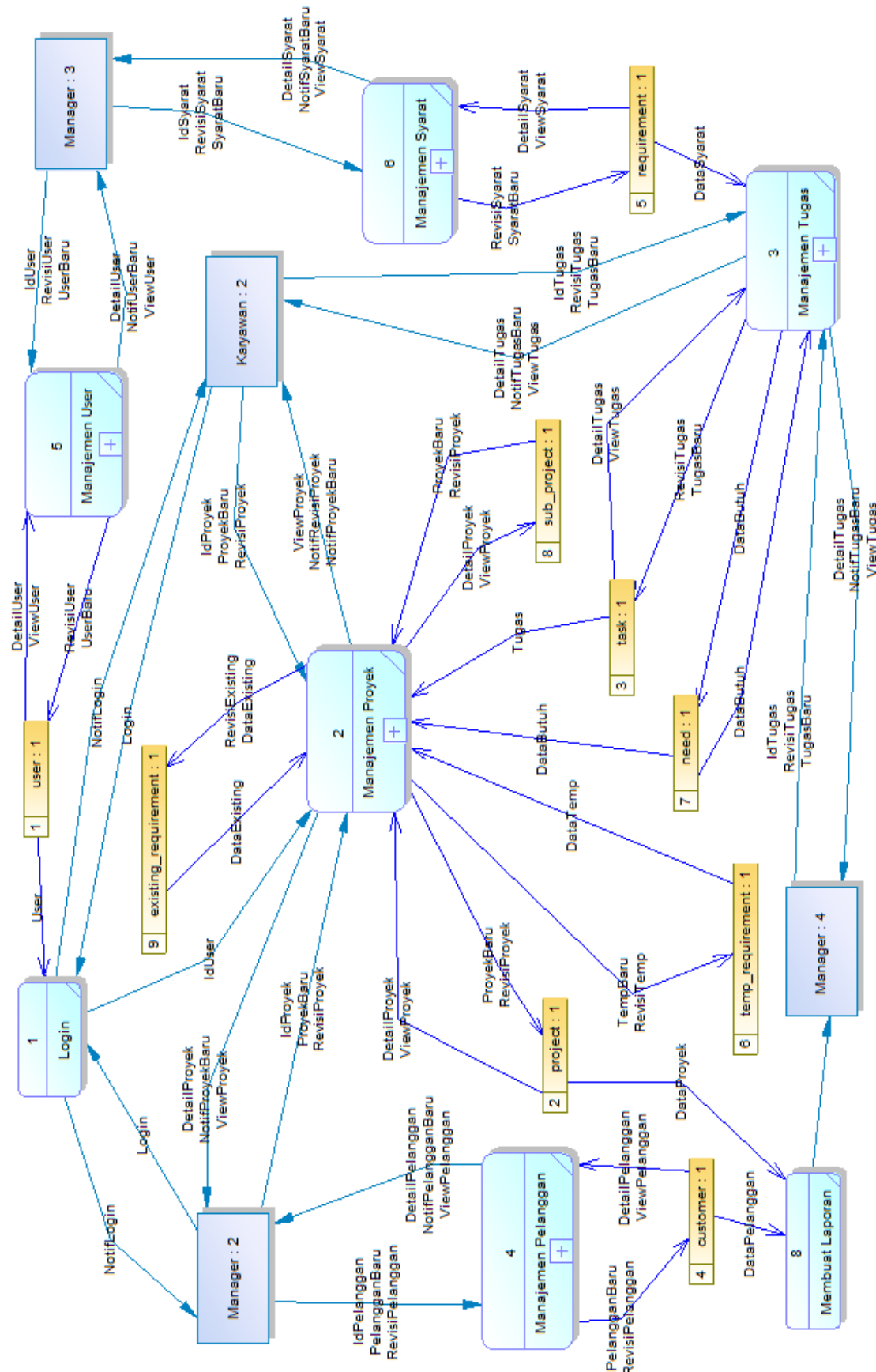
- Adanya pembagian sistem ke dalam sub-sub sistem berdasarkan alur data pada sistem. Adanya *data store* dan alur data (masuk atau keluar) pada sistem. Adanya unsur-unsur eksternal, yaitu sumber dan tujuan dari sistem.
- DFD sebagai alat komunikasi yang baik antara pemakai dengan analis sistem sehingga sangat mudah untuk melakukan kaji ulang secara terus menerus.
- DFD dapat membantu para analis sistem untuk meringkas informasi tentang sistem, mengetahui komponen kunci tentang sistem dan membantu dalam menentukan fungsi-fungsi yang dapat dipakai kembali serta membantu perkembangan aplikasi secara efektif.

Berikut adalah gambar diagram-diagram *Data Flow Diagram* untuk *Task Management System*.

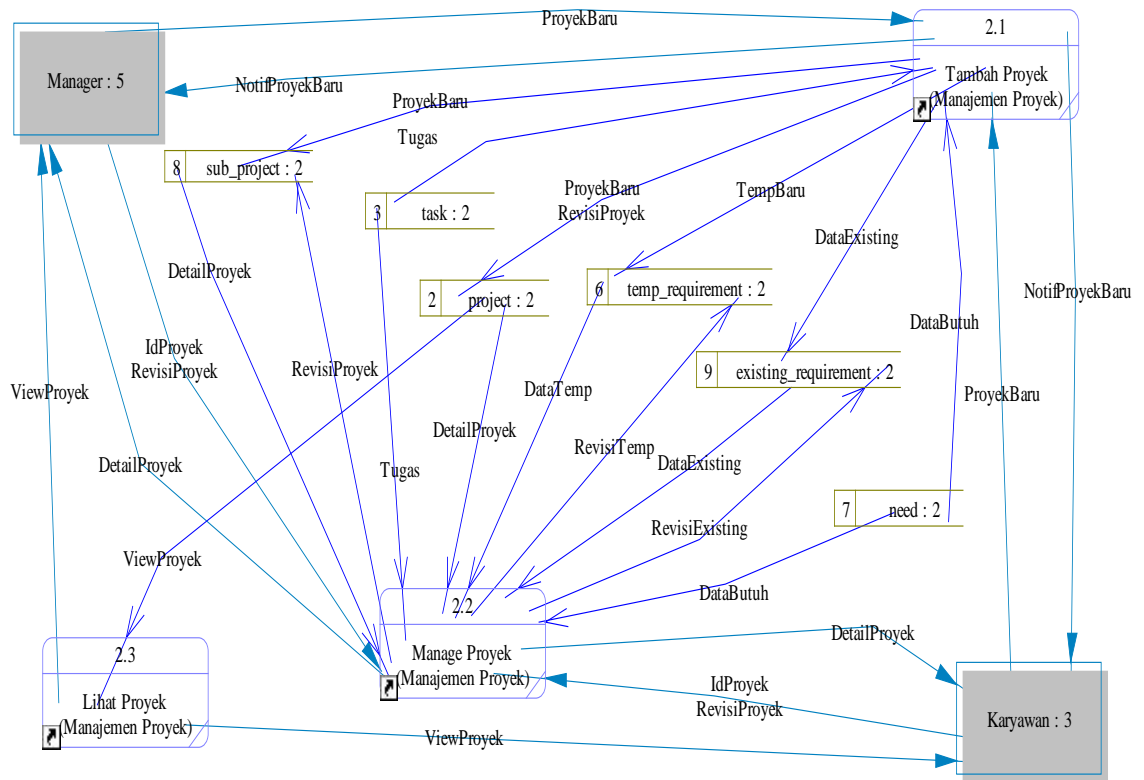


Gambar 3.2 Context Diagram Task Management System

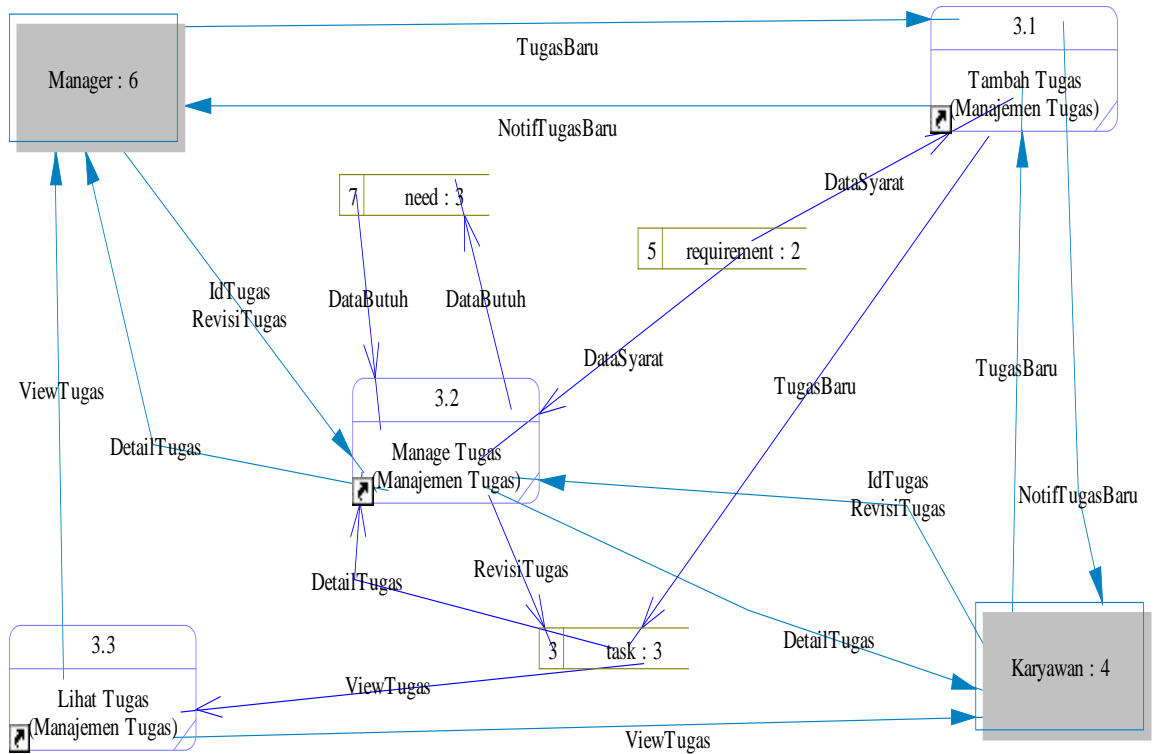




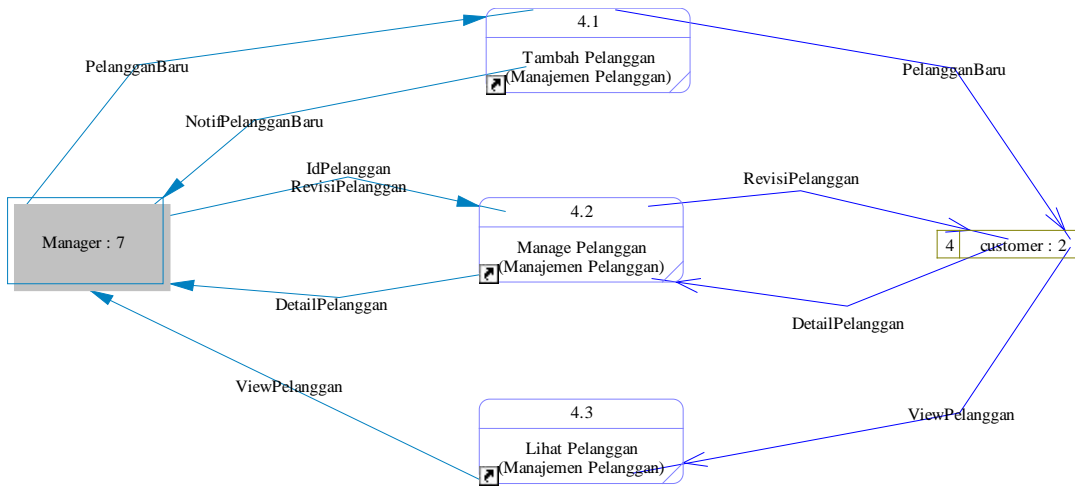
Gambar 3.3 Level-1 Task Management System



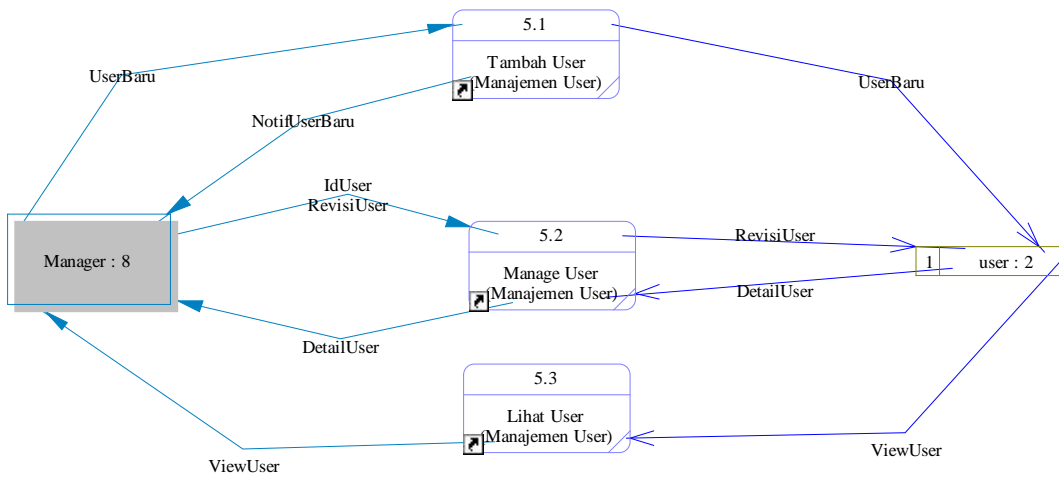
Gambar 3.4 Level-2 Diagram rinci Manajemen Proyek



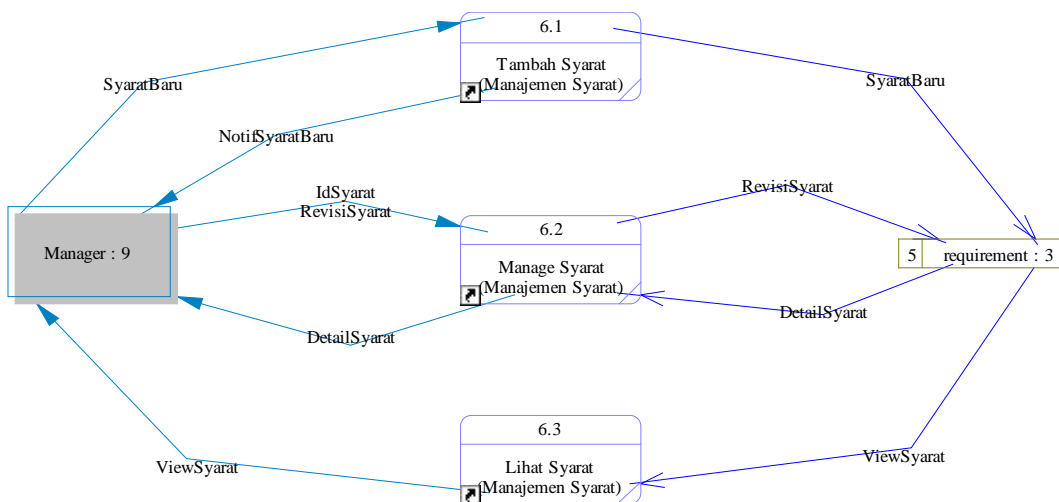
Gambar 3.5 Level-2 Diagram rinci Manajemen Tugas



Gambar 3.6 Level-2 Diagram rinci Manajemen Pelanggan



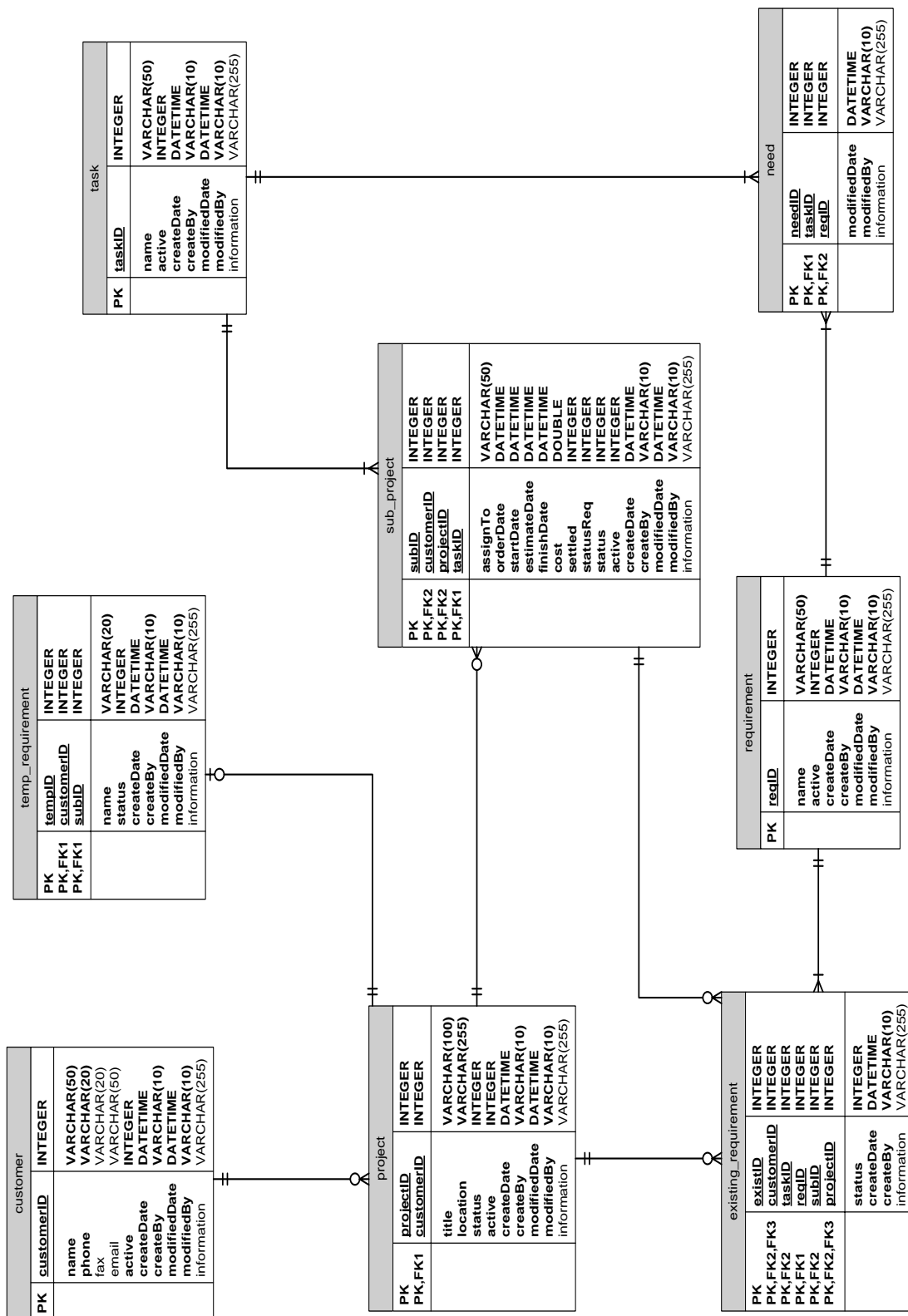
Gambar 3.7 Level-2 Diagram rinci Manajemen User



Gambar 3.8 Level-2 Diagram rinci Manajemen Syarat

## II. *Entity Relationship Diagram*

*Entity Relationship Diagram* (ERD), yaitu sebuah model untuk menjelaskan hubungan antar tabel dalam basis data berdasarkan objek-objek dasar data yang memiliki hubungan antar relasi. Diagram ini digunakan untuk mengkonstruksikan model data konseptual, memodelkan struktur data dan hubungan antar data dan mengimplementasikan basis data secara logika maupun secara fisik. Model data sendiri merupakan sekumpulan cara, peralatan untuk mendeskripsikan data-data yang hubungannya satu sama lain, semantiknya, serta batasan konsistensi. Model data terdiri dari model hubungan entitas dan model relasional. Diagram hubungan entitas ditemukan oleh Peter Chen dalam buku *Entity Relational Model-Toward a Unified of Data*. Dengan diagram ini kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. *Entity Relationship Diagram* dapat membantu dalam menjawab persoalan tentang data yang diperlukan dan bagaimana data tersebut saling berhubungan. Gambar berikut merupakan *Entity Relationship Diagram* dari *Task Management System*.



Gambar 3.9 Entity Relationship Diagram

Uraian hubungan antar entitas dari *Entity Relationship Diagram* sistem ini dapat dilihat pada tabel berikut ini :

Tabel 3.2 Uraian ERD

No	Entity name	Multi- plicity	Relationship	Multi- plicity	Entity name
1	customer	1..1	memiliki/dimiliki	0..N	project
2	user	1..1	memiliki/dimiliki	0..N	reply
3	project	1..1	memiliki/dimiliki	0..N	temp_requirement
4	project	1..1	memiliki/dimiliki	0..N	sub_project
5	project	1..1	memiliki/dimiliki	0..N	existing_requirement
6	sub_project	1..1	memiliki/dimiliki	0..N	reminder
7	sub_project	1..1	memiliki/dimiliki	0..N	existing_requirement
8	requirement	1..1	digunakan/ menggunakan	1..N	existing_requirement
9	requirement	1..1	digunakan/ menggunakan	1..N	need
10	task	1..1	memiliki/dimiliki	1..N	need
11	task	1..1	digunakan/ menggunakan	1..N	sub_project
12	reminder	1..1	memiliki/dimiliki	0..1	reply

### III. Struktur Tabel Database

Dengan menggunakan *Entity Relationship Diagram* yang sudah dibuat, maka dibentuklah database dengan menggunakan *My Structured Query Language (MySQL)*.

*MySQL* adalah sebuah perangkat lunak yang mengimplementasikan sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis.

*MySQL* memiliki beberapa keistimewaan, antara lain :

- **Portabilitas.** *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

- **Perangkat lunak sumber terbuka.** *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- **Multi-user.** *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- **'Performance tuning',** *MySQL* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- **Ragam tipe data.** *MySQL* memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp dan lain-lain.
- **Perintah dan Fungsi.** *MySQL* memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
- **Keamanan.** *MySQL* memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- **Skalabilitas dan Pembatasan.** *MySQL* mampu menangani basis data dalam skala besar, dengan

jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

- **Konektivitas.** *MySQL* dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
- **Lokalisasi.** *MySQL* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
- **Antar Muka.** *MySQL* memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).
- **Klien dan Peralatan.** *MySQL* dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
- **Struktur tabel.** *MySQL* memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.



Dengan menggunakan *MySQL*, maka terbentuklah 9 tabel yang meliputi :

- **Tabel customer**

Nama Tabel : customer

Kegunaan : menyimpan data mengenai pelanggan perusahaan

Primary Key : customerID

Foreign Key : -

Tabel 3.3 Tabel customer

No	Field Name	Type	Length
1	customerID	INTEGER	11
2	name	VARCHAR	50
3	phone	VARCHAR	20
4	fax	VARCHAR	20
5	email	VARCHAR	50
6	active	INTEGER	11
7	createDate	DATETIME	-
8	createBy	VARCHAR	10
9	modifiedDate	DATETIME	-
10	modifiedBy	VARCHAR	10
11	information	VARCHAR	255

- **Tabel user**

Nama Tabel : user

Kegunaan : menyimpan data mengenai user yang akan menggunakan sistem ini

Primary Key : userID

Foreign Key : -

Tabel 3.4 Tabel user

No	Field Name	Type	Length
1	userID	INTEGER	11
2	username	VARCHAR	10
3	password	VARCHAR	20
4	name	VARCHAR	50
5	phone	VARCHAR	20
6	email	VARCHAR	50
7	role	INTEGER	11
8	status	INTEGER	11
9	active	INTEGER	11

Tabel 3.4 Tabel user (lanjutan)

No	Field Name	Type	Length
10	createDate	DATETIME	-
11	createBy	VARCHAR	10
12	modifiedDate	DATETIME	-
13	modifiedBy	VARCHAR	10
14	information	VARCHAR	255

- **Tabel requirement**

Nama Tabel : requirement

Kegunaan : menyimpan syarat-syarat yang diperlukan dalam membuat surat izin

Primary Key : reqID

Foreign Key : -

Tabel 3.5 Tabel requirement

No	Field Name	Type	Length
1	reqID	INTEGER	11
2	name	VARCHAR	50
3	active	INTEGER	11
4	createDate	DATETIME	-
5	createBy	VARCHAR	10
6	modifiedDate	DATETIME	-
7	modifiedBy	VARCHAR	10
8	information	VARCHAR	255

- **Tabel task**

Nama Tabel : task

Kegunaan : Menyimpan data tugas-tugas(surat-surat) yang dapat dikerjakan/dibuat

Primary Key : taskID

Foreign Key : -

Tabel 3.6 Tabel task

No	Field Name	Type	Length
1	taskID	INTEGER	11
2	name	VARCHAR	50
3	active	INTEGER	11
4	createDate	DATETIME	-
5	createBy	VARCHAR	10
6	modifiedDate	DATETIME	-
7	modifiedBy	VARCHAR	10
8	information	VARCHAR	255

- **Tabel need**

- Nama Tabel : need  
 Kegunaan : menghubungkan tabel requirement dan tabel task  
 Primary Key : needID  
 Foreign Key : taskID, reqID

Tabel 3.7 Tabel need

No	Field Name	Type	Length
1	needID	INTEGER	11
2	taskID	INTEGER	11
3	reqID	INTEGER	11
6	modifiedDate	DATETIME	-
7	modifiedBy	VARCHAR	10
8	information	VARCHAR	255

- **Tabel project**

- Nama Tabel : project  
 Kegunaan : menyimpan data mengenai proyek-proyek yang ada di perusahaan  
 Primary Key : projectID  
 Foreign Key : customerID

Tabel 3.8 Tabel project

No	Field Name	Type	Length
1	projectID	INTEGER	11
2	customerID	INTEGER	11
3	title	VARCHAR	100
4	location	VARCHAR	255
5	status	INTEGER	11
6	active	INTEGER	11
7	createDate	DATETIME	-
8	createBy	VARCHAR	10
9	modifiedDate	DATETIME	-
10	modifiedBy	VARCHAR	10
11	information	VARCHAR	255

- **Tabel sub\_project**

- Nama Tabel : sub\_project  
 Kegunaan : menyimpan data mengenai user yang akan menggunakan sistem ini

Primary Key : subID

Foreign Key : -projectID, taskID

Tabel 3.9 Tabel sub\_project

No	Field Name	Type	Length
1	subID	INTEGER	11
2	projectID	INTEGER	10
3	taskID	INTEGER	11
4	assignTo	VARCHAR	50
5	orderDate	DATETIME	-
6	startDate	DATETIME	-
7	estimateDate	DATETIME	-
8	finishDate	DATETIME	-
9	cost	DOUBLE	-
10	settled	INTEGER	11
11	statusReq	INTEGER	11
12	status	INTEGER	11
13	active	INTEGER	11
14	createDate	DATETIME	-
15	createBy	VARCHAR	10
16	modifiedDate	DATETIME	-
17	modifiedBy	VARCHAR	10
18	information	VARCHAR	255

- **Tabel existing\_requirement**

Nama Tabel : existing\_requirement

Kegunaan : untuk mengetahui syarat-syarat/  
tugas-tugas yang sudah ada/selesai  
dikerjakan

Primary Key : existID

Foreign Key : projectID, subID, reqID

Tabel 3.10 Tabel existing\_requirement

No	Field Name	Type	Length
1	existID	INTEGER	11
2	projectID	INTEGER	11
3	subID	INTEGER	11
4	reqID	INTEGER	11
5	status	INTEGER	11
6	createDate	DATETIME	-
7	createBy	VARCHAR	10
8	information	VARCHAR	255

- **Tabel temp\_requirement**

Nama Tabel : temp\_requirement

Kegunaan : untuk mengetahui syarat-syarat  
tambahan untuk mengerjakan  
sebuah tugas/sub-proyek

Primary Key : tempID

Foreign Key : subID

Tabel 3.11 Tabel temp\_requirement

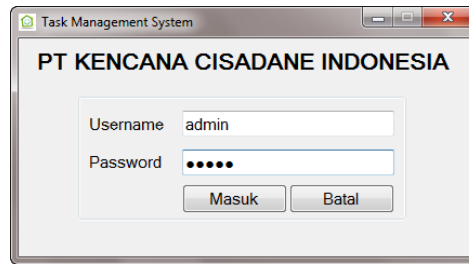
No	Field Name	Type	Length
1	tempID	INTEGER	11
2	subID	INTEGER	11
3	name	VARCHAR	20
4	createDate	DATETIME	-
5	createBy	VARCHAR	10
6	modifiedDate	DATETIME	-
7	modifiedBy	VARCHAR	10
8	information	VARCHAR	255

#### IV. Desain *Interface*

Setelah desain struktur dibuat, barulah masuk ke tahap desain interface yaitu dengan membuat tampilan tiap-tiap halaman yang akan dibuat di dalam aplikasi. Tampilan-tampilan umum yang dibuat yaitu :

- Halaman Login

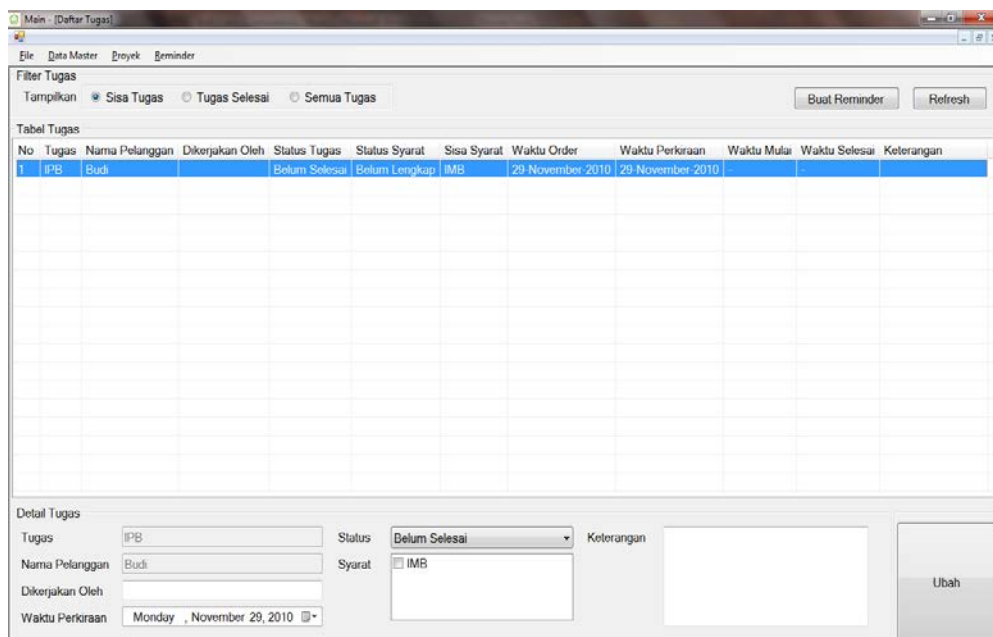
Pada halaman ini user diminta untuk mengisi username dan password untuk masuk ke dalam sistem. Terdapat 2 jenis user yang memiliki hak akses yang berbeda, yaitu user untuk pegawai dan user untuk admin. Jika user memasukkan username atau password yang salah, maka akan muncul peringatan. Jika benar, maka akan muncul konfirmasi bahwa user berhasil masuk ke dalam sistem.



Gambar 3.10 Halaman Login

- Halaman Main (Daftar Tugas)

Pada halaman ini, user dapat melihat daftar tugas yang dapat disaring berdasarkan status tugasnya yaitu sisa tugas, tugas selesai dan semua tugas. User juga dapat mengubah data tugas-tugas yang sudah ada di dalam sistem.

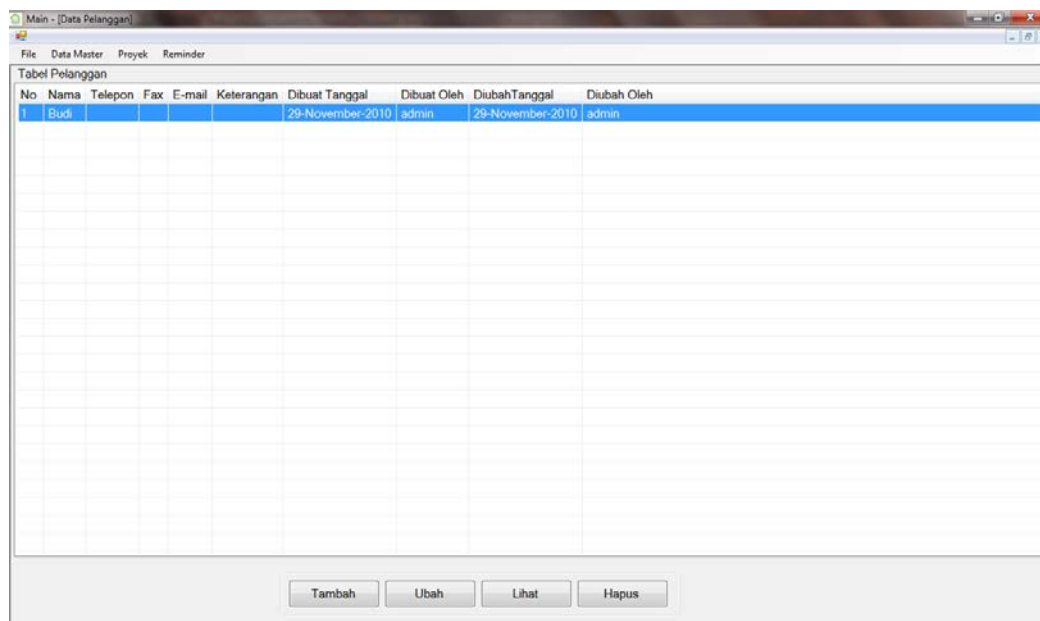


Gambar 3.11 Halaman Main (Daftar Tugas)

- Halaman data pelanggan

Pada halaman data pelanggan, user dapat menambah pelanggan, mengubah data pelanggan, melihat data

pelanggan serta menghapus pelanggan. Jika user memilih untuk menghapus pelanggan, maka akan muncul konfirmasi untuk menghapus pelanggan sebelum pelanggan dihapus dari sistem.



Gambar 3.12 Halaman Data Pelanggan

- Halaman tambah, ubah, lihat pelanggan

Halaman tambah, ubah dan lihat pelanggan memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah pelanggan, user dapat menambah pelanggan baru. Pada halaman ubah pelanggan, user dapat mengubah data pelanggan yang dipilih. Sedangkan pada halaman lihat pelanggan, user hanya dapat melihat data pelanggan.

Tambah Pelanggan

Pelanggan

Nama

Telepon

Fax

Email

Keterangan

Dibuat Tanggal 15-December-2010 02:53:05

Dibuat Oleh admin

Diubah Tanggal 15-December-2010 02:53:05

Diubah Oleh admin

Simpan Reset Batal

Gambar 3.13 Halaman Tambah Pelanggan

Ubah Pelanggan

Pelanggan

Nama

Telepon

Fax

Email

Keterangan

Dibuat Tanggal 29-November-2010 16:15:02

Dibuat Oleh admin

Diubah Tanggal 15-December-2010 02:53:35

Diubah Oleh admin

Simpan Reset Batal

Gambar 3.14 Halaman Ubah Pelanggan

Lihat Pelanggan

Pelanggan

Nama

Telepon

Fax

Email

Keterangan

Dibuat Tanggal 29-November-2010 16:15:02

Dibuat Oleh admin

Diubah Tanggal 29-November-2010 16:15:02

Diubah Oleh admin

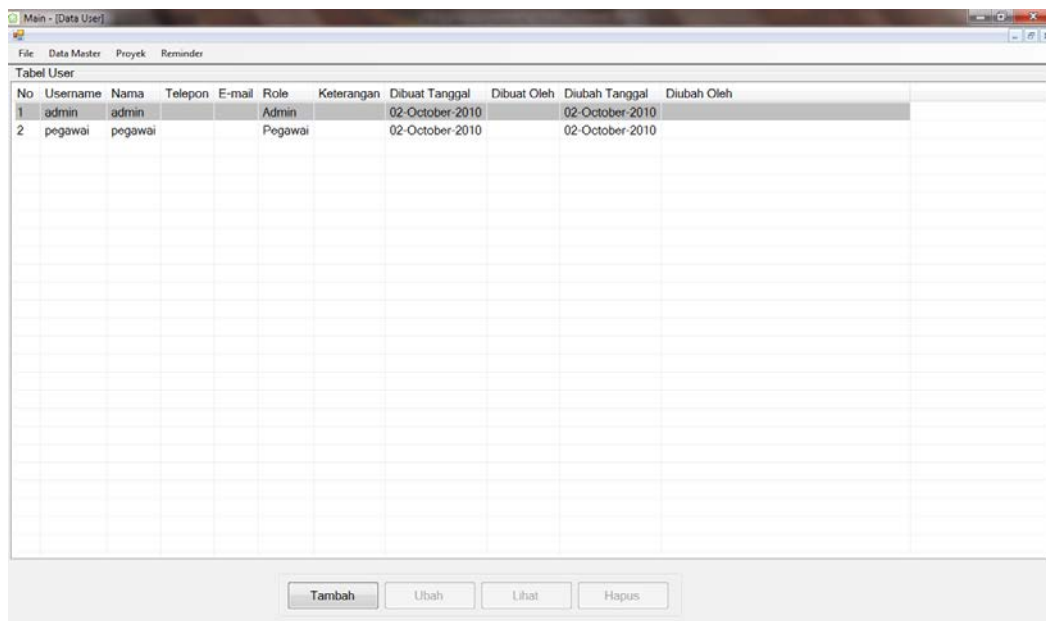
Simpan Reset Batal

Gambar 3.15 Halaman Lihat Pelanggan



- Halaman data user

Pada halaman data user, user dapat menambah user, mengubah data user, melihat data user serta menghapus user. Jika user memilih untuk menghapus user, maka akan muncul konfirmasi untuk menghapus user sebelum user dihapus dari sistem.



No	Username	Nama	Telepon	E-mail	Role	Keterangan	Dibuat Tanggal	Dibuat Oleh	Diubah Tanggal	Diubah Oleh
1	admin	admin			Admin		02-October-2010		02-October-2010	
2	pegawai	pegawai			Pegawai		02-October-2010		02-October-2010	

Gambar 3.16 Halaman Data User

- Halaman tambah, ubah, lihat user

Halaman tambah, ubah dan lihat user memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah user, user dapat menambah user baru. Pada halaman ubah user, user dapat mengubah data user yang dipilih. Sedangkan pada halaman lihat user, user hanya dapat melihat data user.

Tambah User

User

Nama

Username

Password

Telepon

Role

Email

Keterangan

Dibuat Tanggal

Dibuat Oleh

Diubah Tanggal

Diubah Oleh

Gambar 3.17 Halaman Tambah User

Ubah User

User

Nama

Username

Password

Telepon

Role

Email

Keterangan

Dibuat Tanggal

Dibuat Oleh

Diubah Tanggal

Diubah Oleh

Gambar 3.18 Halaman Ubah User

Lihat User

User

Nama

Username

Password

Telepon

Role

Email

Keterangan

Dibuat Tanggal

Dibuat Oleh

Diubah Tanggal

Diubah Oleh

Gambar 3.19 Halaman Lihat User

- Halaman data tugas (surat izin)

Pada halaman data tugas (surat izin), user dapat menambah tugas yang dapat dikerjakan, mengubah data tugas, melihat data tugas serta menghapus tugas. Jika user memilih untuk menghapus tugas, maka akan muncul konfirmasi untuk menghapus tugas sebelum tugas dihapus dari sistem.

No	Nama Tugas	Butuh Syarat	Keterangan	Dibuat Tanggal	Dibuat Oleh	Diubah Tanggal	Diubah Oleh
1	IPB	STNK, IMB	Keterangan	29-November-2010	admin	29-November-2010	admin

Gambar 3.20 Halaman Data Tugas

- Halaman tambah, ubah, lihat tugas

Halaman tambah, ubah dan lihat tugas memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah tugas, user dapat menambah tugas baru. Pada halaman ubah tugas, user dapat mengubah data tugas yang dipilih. Sedangkan pada halaman lihat tugas, user hanya dapat melihat data tugas.

Tambah Tugas

Tugas

Nama

Butuh Syarat

- KTP
- STNK
- IMB

Keterangan

Dibuat Tanggal: 15-December-2010 03:17:53

Dibuat Oleh: admin

Diubah Tanggal: 15-December-2010 03:17:53

Diubah Oleh: admin

Simpan Reset Batal

Gambar 3.21 Halaman Tambah Tugas

Ubah Tugas

Tugas

Nama: IPB

Butuh Syarat

- KTP
- STNK
- IMB

Keterangan

Dibuat Tanggal: 29-November-2010 16:13:39

Dibuat Oleh: admin

Diubah Tanggal: 15-December-2010 03:18:52

Diubah Oleh: admin

Simpan Reset Batal

Gambar 3.22 Halaman Ubah Tugas

Lihat Tugas

Tugas

Nama: IPB

Butuh Syarat

- STNK
- IMB

Keterangan

Dibuat Tanggal: 29-November-2010 16:13:39

Dibuat Oleh: admin

Diubah Tanggal: 29-November-2010 16:13:39

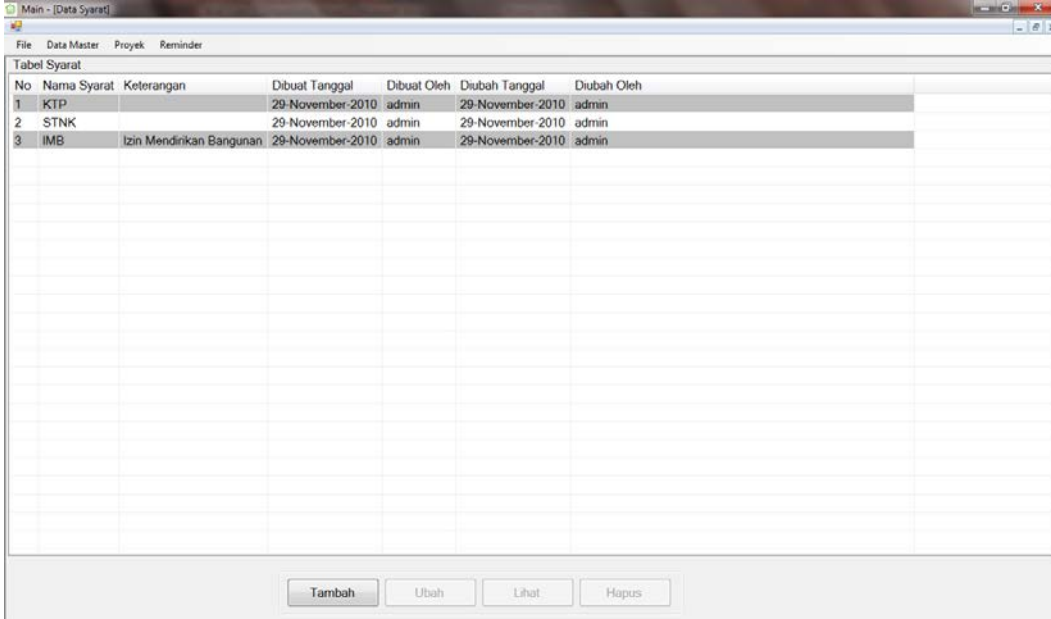
Diubah Oleh: admin

Simpan Reset Batal

Gambar 3.23 Halaman Lihat Tugas

- Halaman data syarat

Pada halaman data syarat, user dapat menambah syarat-syarat yang diperlukan sebelum mengerjakan tugas (surat izin), mengubah data syarat, melihat data syarat serta menghapus syarat. Jika user memilih untuk menghapus syarat, maka akan muncul konfirmasi untuk menghapus syarat sebelum syarat dihapus dari sistem.



No	Nama Syarat	Keterangan	Dibuat Tanggal	Dibuat Oleh	Diubah Tanggal	Diubah Oleh
1	KTP		29-November-2010	admin	29-November-2010	admin
2	STNK		29-November-2010	admin	29-November-2010	admin
3	IMB	Izin Mendirikan Bangunan	29-November-2010	admin	29-November-2010	admin

Gambar 3.24 Halaman Data Syarat

- Halaman tambah, ubah, lihat syarat

Halaman tambah, ubah dan lihat syarat memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah syarat, user dapat menambah syarat baru. Pada halaman ubah syarat, user dapat mengubah data syarat yang dipilih. Sedangkan pada halaman lihat syarat, user hanya dapat melihat data syarat.

Tambah Syarat

Syarat

Nama Syarat

Keterangan

Dibuat Tanggal 15-December-2010 03:23:13

Dibuat Oleh admin

Diubah Tanggal 15-December-2010 03:23:13

Diubah Oleh admin

Simpan Reset Batal

Gambar 3.25 Halaman Tambah Syarat

Ubah Syarat

Syarat

Nama Syarat

Keterangan

Dibuat Tanggal 29-November-2010 16:11:58

Dibuat Oleh admin

Diubah Tanggal 15-December-2010 03:23:37

Diubah Oleh admin

Simpan Reset Batal

Gambar 3.26 Halaman Ubah Syarat

Lihat Syarat

Syarat

Nama Syarat

Keterangan

Dibuat Tanggal 29-November-2010 16:11:58

Dibuat Oleh admin

Diubah Tanggal 29-November-2010 16:12:18

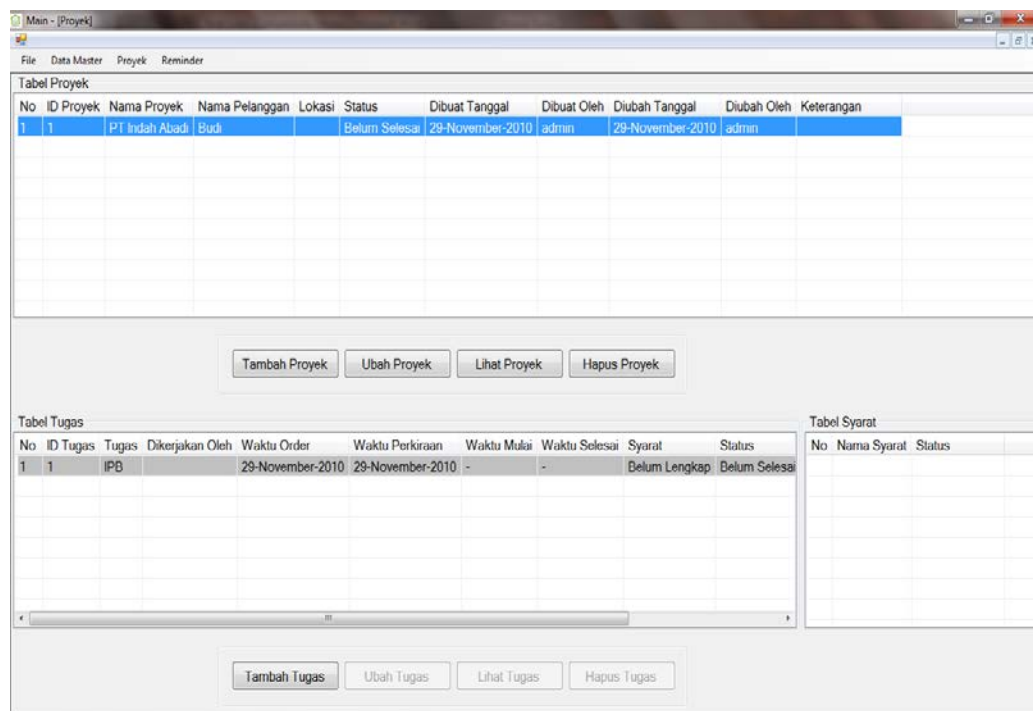
Diubah Oleh admin

Simpan Reset Batal

Gambar 3.27 Halaman Lihat Syarat

- Halaman proyek

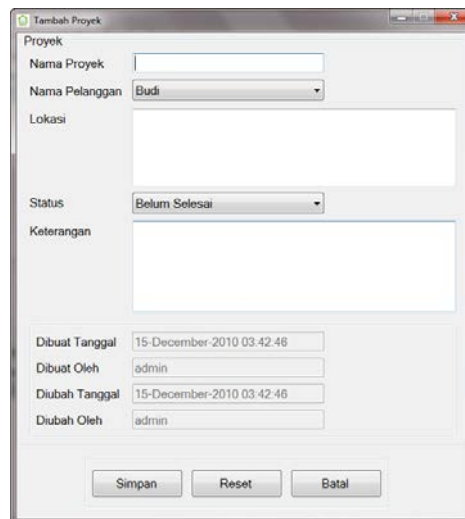
Pada halaman proyek, user dapat menambah proyek, mengubah proyek, melihat proyek, menghapus proyek. Bila sudah terdapat proyek di dalam sistem, barulah user dapat menambah tugas yang akan dikerjakan, mengubah data tugas yang akan dikerjakan, melihat data tugas yang akan dikerjakan, serta menghapus tugas yang akan dikerjakan. Pada halaman ini user juga dapat melihat syarat-syarat yang diperlukan untuk dapat menyelesaikan sebuah tugas di dalam sebuah proyek. Jika user memilih untuk menghapus proyek/tugas, maka akan muncul konfirmasi untuk menghapus proyek/tugas sebelum proyek/tugas dihapus dari sistem.



Gambar 3.28 Halaman Proyek

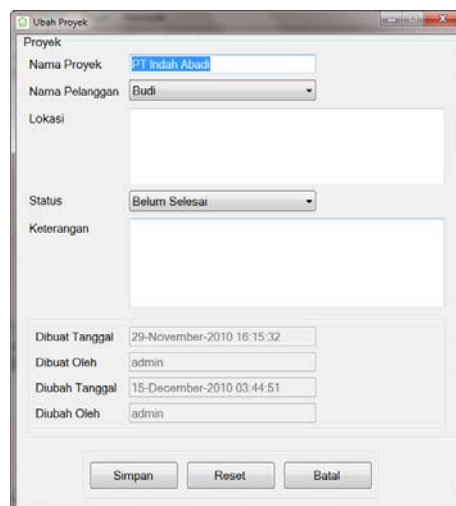
- Halaman tambah, ubah, lihat proyek

Halaman tambah, ubah dan lihat proyek memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah proyek, user dapat menambah proyek baru. Pada halaman ubah proyek, user dapat mengubah data proyek yang dipilih. Sedangkan pada halaman lihat proyek, user hanya dapat melihat data proyek.



The screenshot shows a web form titled "Tambah Proyek". It contains several input fields and dropdown menus. The "Nama Pelanggan" dropdown is set to "Budi". The "Status" dropdown is set to "Belum Selesai". The "Dibat Tanggal" and "Diubah Tanggal" fields show the date and time "15-December-2010 03:42:46". The "Dibat Oleh" and "Diubah Oleh" fields show the name "admin". At the bottom, there are three buttons: "Simpan", "Reset", and "Batal".

Gambar 3.29 Halaman Tambah Proyek



The screenshot shows a web form titled "Ubah Proyek". It contains several input fields and dropdown menus. The "Nama Proyek" field is highlighted in blue and contains the text "PT Indah Abadi". The "Nama Pelanggan" dropdown is set to "Budi". The "Status" dropdown is set to "Belum Selesai". The "Dibat Tanggal" field shows the date and time "29-November-2010 16:15:32". The "Dibat Oleh" field shows the name "admin". The "Diubah Tanggal" field shows the date and time "15-December-2010 03:44:51". The "Diubah Oleh" field shows the name "admin". At the bottom, there are three buttons: "Simpan", "Reset", and "Batal".

Gambar 3.30 Halaman Ubah Proyek



Gambar 3.31 Halaman Lihat Proyek

- Halaman tambah, ubah, lihat tugas (sub-proyek)

Halaman tambah, ubah dan lihat tugas (sub-proyek) memiliki tampilan yang sama, namun memiliki fungsi dan *behavior* (tingkah laku) yang berbeda. Pada halaman tambah tugas (sub-proyek), user dapat menambah tugas (sub-proyek) baru. Pada halaman ubah tugas (sub-proyek), user dapat mengubah data tugas (sub-proyek) yang dipilih. Sedangkan pada halaman lihat tugas (sub-proyek), user hanya dapat melihat data tugas (sub-proyek).

Gambar 3.32 Halaman Tambah Tugas (sub-proyek)

Gambar 3.33 Halaman Ubah Tugas (sub-proyek)

Gambar 3.34 Halaman Lihat Tugas (sub-proyek)

### C. Coding dan debugging

Tahap berikutnya adalah coding yaitu menulis sejumlah instruksi yang membentuk suatu fungsi yang diperlukan sesuai dengan spesifikasi modul yang telah dibuat sebelumnya dan

menggabungkannya dengan interface yang sudah dibuat sebelumnya. Coding dilakukan dengan menggunakan, antara lain:

- *.NET Framework*

*.NET Framework* merupakan sebuah komponen yang dapat ditambahkan ke sistem operasi Microsoft Windows atau telah terintegrasi ke dalam Windows. Kerangka kerja ini menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk framework ini. *.NET* seringkali juga dapat diartikan sebagai platform, yang merupakan suatu lingkungan terpadu untuk pengembangan dan eksekusi untuk berbagai macam bahasa pemrograman dan kumpulan *library* untuk bekerja sama membuat dan menjalankan aplikasi berbasis Windows yang lebih mudah untuk dibuat, diatur, didistribusikan dan diintegrasikan dengan sistem jaringan lain.

- Bahasa pemrograman *C#* (dibaca *C-sharp*)

*C#* merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa *C++* yang telah dipengaruhi oleh aspek-aspek ataupun

fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya dengan beberapa penyerdehanaan. Beberapa keunggulan C# yang membuat penulis menggunakan bahasa pemrograman ini, yaitu :

- C# dibuat sebagai bahasa pemrograman yang bersifat bahasa pemrograman *general-purpose* (untuk tujuan jamak), berorientasi objek, modern dan sederhana.
- C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi.
- C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server maupun sistem embedded, mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi.
- Bahasa C# memiliki *type checking* yang kuat, pengecekan larik (*array*), pendektesian terhadap percobaan terhadap penggunaan variabel-variabel yang belum diinisialisasikan,

portabilitas *source code* dan *garbage collection* secara otomatis.

- *NHibernate*

*NHibernate* adalah sebuah solusi Object-Relational Mapping (ORM) untuk *.NET Framework*. ORM merupakan sebuah teknik yang digunakan dalam pemrograman untuk menggunakan basis data relasional sebagai penyimpanan data dengan bentuk objek. Teknik ini biasa digunakan dalam bahasa pemrograman berorientasi objek saat harus menggunakan basis data relasional dalam penyimpanannya. Keunggulan dari penggunaan *NHibernate* adalah kita dapat membebaskan diri dari *query* terhadap tabel dimana untuk fungsi insert, update atau delete dimungkinkan untuk sama sekali tidak menulis *query* sehingga membuat kode lebih sedikit (mengefisienkan kode).

Coding dimulai dari modul *Master* dan kemudian modul *Project*. Fungsi-fungsi yang decoding pada tiap-tiap modul secara umum meliputi fungsi untuk menambah data (tambah), mengubah data (ubah), menghapus data (hapus), melihat data (lihat), memfilter data dan mempopulasikan data. Pada tahap ini juga dilakukan debugging yaitu mencari dan mengurangi *bug* atau kerusakan di dalam sebuah program komputer. Debugging dilakukan setiap kali sebuah fungsi telah selesai dibuat. Hal ini

dilakukan untuk mengetahui *bug* lebih awal sebelum *bug-bug* tersebut menumpuk dan sulit untuk diatasi.

#### **D. Testing dan Implementasi**

Setelah semua fungsi-fungsi yang sudah diintegrasikan menjadi sebuah perangkat lunak berjalan lancar, maka dilakukan testing untuk memeriksa apakah perangkat lunak tersebut berjalan lancar sesuai harapan dan bebas dari *bug* maupun kerusakan program yang tidak diinginkan. Testing dilakukan dengan melakukan *Integration Testing*, yaitu menguji integrasi yang dilakukan kepada program seperti halnya ketika program tersebut telah diinstal di komputer klien yang membutuhkan integrasi dengan sistem lain, seperti halnya integrasi dengan database, kemudian *Validation Testing* yaitu menguji masukan yang diberikan kepada program dan memeriksa apakah keluarannya sesuai dengan yang diinginkan dan yang terakhir adalah *System Testing* yaitu dengan menguji performa dari program pada kondisi tertentu.

Setelah tahap testing berhasil dilalui, maka dilakukan implementasi perangkat lunak yang telah dibuat dengan menginstal perangkat lunak tersebut ke komputer-komputer perusahaan, kemudian melakukan testing untuk menguji apakah perangkat lunak tersebut dapat berjalan lancar di masing-masing komputer dan yang terakhir memberikan *training* kepada user-user yang akan menggunakan komputer tersebut.

### **3.3.2 Kendala yang Ditemukan**

Selama kerja magang di PT. Kencana Cisadane Indonesia, terdapat beberapa kendala. Sebagian besar kendala tersebut adalah sulitnya untuk memahami keinginan user akan desain dari interface serta fungsi pada tiap modul. Selain itu, Manager Operasional yang mengkoordinasi penulis sering keluar kantor oleh karena kesibukan di luar kantor sehingga sulit untuk bertemu dan mendiskusikan perangkat lunak yang dibuat.

Namun kendala-kendala tersebut tidak terlalu menghambat pada proses pembuat perangkat lunak tersebut karena penulis menemukan dan melakukan beberapa solusi seperti yang akan dijelaskan pada sub bab selanjutnya.

### **3.3.3 Solusi Atas Kendala yang Ditemukan**

Solusi yang dilakukan untuk mengatasi kendala dalam memahami user pada bagian interface dan fungsi pada modul adalah dengan melakukan analysis requirement dan functional specification yang lebih detail. Serta dengan melakukan presentasi setiap kali modul, fungsi atau interface selesai dibuat, sebelum semua modul diintegrasikan.

Dengan begitu user dapat melakukan review kekurangan dari modul tersebut dan penulis melakukan revisi pada modul tersebut sebelum terlambat. Sehingga interface dan fungsi dari

perangkat lunak tersebut dapat sesuai dengan keinginan user. Selain itu, setiap minggunya akan dilakukan presentasi atau diskusi dengan Manager Operasional mengingat kesibukan beliau di luar kantor sehingga beliau mengetahui perkembangan perangkat lunak yang sedang dibuat dan dapat merevisi jika diperlukan.