

# PENGEMBANGAN APLIKASI MANAJEMEN PROYEK SECARA GRAFIS BERBASIS WEB MENGGUNAKAN METODE DIAGRAM PRESEDENSI

Suhadi Lili, Yudhi Purwananto, Aris Nurahman Nihayanto

Jurusan Teknik Informatika,

Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804

Email: suhadi@its-sby.edu, yudhi@its-sby.edu

## ABSTRAK

*Dalam kegiatan manajemen proyek diperlukan koordinasi yang bagus dan kerjasama tim yang kompak antar divisi pelaksana proyek. Selama ini, secara konvensional untuk membuat perencanaan proyek maupun melakukan review dan evaluasi jalannya proyek, perlu diadakan rapat atau pertemuan langsung dari orang-orang yang berperan sebagai manajer dan perencana. Para manajer profesional ini seringkali adalah orang-orang yang mobilitasnya tinggi sehingga tidak selalu berada ditempat pada saat dibutuhkan, sehingga koordinasi proyek akan terhambat dan menjadi tidak efisien. Dengan adanya kemajuan teknologi pada saat ini khususnya internet, permasalahan tersebut sangat memungkinkan untuk diselesaikan dengan menyediakan perangkat lunak yang dikhususkan untuk manajemen proyek, yang bisa diakses online secara multiuser dengan pendekatan sifat real time sehingga para manajer dapat melakukan kolaborasi kerja di mana saja mereka berada selama tersedia perangkat komputer yang terhubung dengan internet.*

*Penelitian ini bertujuan untuk membuat perangkat lunak tersebut. Dalam Teknik Informatika, aplikasi seperti ini merupakan aplikasi terdistribusi, karena itu idealnya harus dibuat bersifat portabel agar dapat berjalan pada semua sistem operasi komputer, dan tidak terhambat oleh restriksi internet (sering kali berupa firewall). Untuk mewujudkan sifat-sifat tersebut, aplikasi ini disusun dengan teknologi J2EE, Java, Web Service, dan untuk enkapsulasi data digunakan struktur XML karena fleksibel dan mampu merepresentasikan struktur hirarki data berbentuk tree. Dalam perancangan aplikasi ini digunakan bahasa UML untuk memodelkan aplikasi, sedangkan pemodelan data menggunakan Conceptual Data Model dan Physical Data Model*

*Fitur utama yang dibuat untuk manajemen proyek pada aplikasi ini adalah komponen Gantt Chart yang menerapkan metode Diagram Presedensi. Komponen ini dapat membantu manajer untuk melakukan manajemen proyek dengan menyediakan fitur untuk menggambarkan secara visual pelaksanaan proyek dengan elemen-elemen Gantt Chart yaitu : Task Sheet, Diagram Pane, Time Line, dan Work Calendar. Dengan penerapan metode Diagram Presedensi akan mempermudah perencana proyek dalam penentuan jadwal start dan finish dari task-task penyusun proyek Aplikasi yang telah dibuat telah mengalami uji coba untuk berjalan secara multiuser, dan dari beberapa kali skenario uji coba yang dijalankan aplikasi sudah berjalan sesuai dengan spesifikasi yang telah ditetapkan.*

**Kata Kunci :** *Project Management, Gantt Chart, Diagram Presedensi*

## 1. PENDAHULUAN

Dengan adanya perkembangan teknologi dan era globalisasi yang tengah bergulir, banyak sekali lahan bisnis dan pekerjaan baru bermunculan. Sebagian dari bisnis dan pekerjaan ini membutuhkan mobilitas yang sangat tinggi dari para pelaku bisnisnya meliputi manajer proyek atau bisnis, *line manager* perusahaan, karyawan, dan juga *customer*. Contoh yang dapat kita ambil misalnya adalah sebuah proyek pembuatan software untuk mengimplementasikan proses bisnis sebuah perusahaan atau industri yang dikerjakan oleh perusahaan pembuat software.

Kebanyakan *customer* ini berada pada tempat yang berjauhan, dengan kemungkinan tiap *customer* juga memiliki unit-unit bisnis yang berjauhan, sehingga seorang manajer proyek software, analisis sistem, maupun pemrogram software harus memiliki mobilitas tinggi untuk berhubungan dengan *customer* seperti ini.

Untuk melakukan manajemen sebuah proyek seperti itu, tentunya diperlukan sebuah perangkat pendukung yang mampu dijadikan sebagai sarana komunikasi dan interaksi antar *participant* (peserta) proyek, yaitu sebuah sistem perangkat lunak

manajemen proyek yang sifatnya online, dan mudah diakses dari mana saja. Akan lebih baik lagi apabila perangkat lunak ini mempunyai *visual interface* yang interaktif dan fleksibel sehingga memberikan dukungan yang sangat signifikan terhadap penyampaian informasi.

Dengan adanya aplikasi manajemen proyek online, efisiensi waktu dan biaya perusahaan dapat ditingkatkan. Hal ini bisa dicapai karena masing-masing peserta proyek yaitu :

1. *Manajer Proyek*, baik perorangan maupun sebuah tim, dapat membuat sebuah perencanaan proyek baik penjadwalan maupun alokasi sumber daya perusahaan dari mana saja ia berada, karena umumnya manajer proyek ini adalah individu yang memiliki mobilitas yang sangat tinggi. Ia juga dapat menyesuaikan rencana proyek, tergantung dari kelancaran jalannya proyek.
2. *Karyawan Proyek*, dapat memperoleh penugasan pekerjaan kapan saja, serta dapat melaporkan perkembangan pekerjaan yang dilakukannya, umumnya adalah berupa persentase kemajuan (*progress*).
3. *Stakeholder Proyek*, yaitu siapapun baik perorangan maupun organisasi yang berkepentingan dengan proyek, dapat memantau dan memperoleh informasi tentang jalannya proyek.

Beberapa permasalahan yang harus diselesaikan dalam pembuatan aplikasi *project management online* dilatarbelakangi oleh keinginan untuk mewujudkan tentang aplikasi terdistribusi yang paling ideal, agar hasilnya dapat memberikan performa yang optimal, yaitu :

1. Bagaimana membuat aplikasi *project management* khususnya fasilitas *Gantt Chart* yang merupakan komponen visual dengan elemen *Task Sheet*, *Diagram Pane*, *Time Line*, dan *Work Calendar* dengan pemrograman Java agar bersifat portabel.
2. Bagaimana menerapkan metode Diagram Presedensi terhadap *task-task* yang menyusun *Gantt Chart* untuk mempermudah manajemen proyek.
3. Bagaimana merancang dan mengatur mekanisme agar aplikasi *project management* dapat digunakan secara *online* dan *multiuser* dengan pendekatan sifat *real time*.
4. Bagaimana menerapkan sistim komunikasi data berbasis *Web Service* pada aplikasi *project management*, dengan format data menggunakan XML, agar aplikasi dapat diakses dengan koneksi internet dengan hambatan *firewall* seminimal mungkin sesuai sifat web.
5. Bagaimana merancang aplikasi *project management* dalam arsitektur *three-tier*, yaitu dibagi dalam *Presentation Layer*, *Business Logic Layer* (*middle-tier*), dan *Data Access Layer*

karena dengan arsitektur ini akan mudah dilakukan pengembangan dan *deployment*.

6. Karena merupakan sebuah aplikasi *case-tool* visual, harus dirancang bagaimana supaya interface *client* bisa dibuat interaktif sehingga akan membuat kegiatan manajemen proyek menjadi efektif.

## 2. MANAJEMEN PROYEK, J2EE, DAN WEB SERVICE

Manajemen proyek dikatakan bagus bila dapat mengoptimalkan tiga faktor yaitu *waktu*, *biaya*, dan *performa* atau *kualitas*. Sebuah proyek dikatakan sukses apabila diselesaikan tepat waktu, dalam anggaran yang telah ditentukan, dan menunjukkan performa atau kualitas seperti yang ditetapkan sebelumnya. Terdapat beberapa teknik, metodologi dan alat (*tools*) untuk melakukan manajemen proyek, yaitu [8]:

### 1. Work Breakdown Structure (WBS)

WBS berupa struktur hirarki yang digunakan untuk mengorganisasi *task* (pekerjaan) untuk membuat *schedule report* dan melakukan *track* terhadap biaya.

### 2. Gantt Chart

Representasi secara visual grafis yang digunakan untuk menginformasikan sebuah proyek. Umumnya sebuah *Gantt Chart* terdiri dari sebuah *sheet* yang berisi daftar *task* dan sebuah panel diagram dimana *task* direpresentasikan dalam bentuk sebuah bar horizontal.

### 3. PERT / CPM (Critical Path Method)

Kedua metode tersebut menggunakan representasi grafis untuk menggambarkan sebuah proyek yang disebut *Project Network* atau *CPM Diagram* untuk menunjukkan inter-relasi antar element dari sebuah proyek dan menunjukkan urutan-urutan aktivitas yang harus dijalankan.

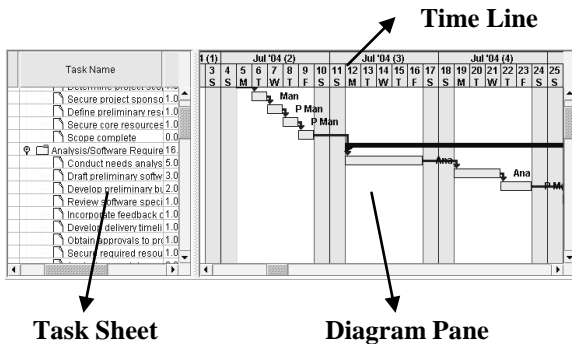
Dalam penelitian ini digunakan metode kedua, yaitu menggunakan Gantt Chart.

### Gantt Chart

*Gantt Chart* adalah sebuah *tool* pembantu yang dapat digunakan untuk melakukan manajemen proyek dalam bentuk diagram. Implementasi *Gantt Chart*, dalam prakteknya mempunyai versi yang bermacam-macam antar vendor aplikasi, mulai dari yang paling sederhana seperti yang diperkenalkan oleh penemunya yaitu Henry L. Gantt, hingga ditambahkannya bermacam-macam fitur seperti misalnya Presedensi *Task*. Secara umum *Gantt Chart* dibagi menjadi 4 unsur yaitu *Task Sheet*, *Time Line*, *Diagram Pane* dan *Work Calendar*. *Work Calendar* adalah kalender kerja pelaksanaan proyek yang secara visual tampak dalam *Diagram Pane* sebagai garis-garis vertikal yang menunjukkan hari kerja aktif

dan hari libur. *Work Calendar* terdiri dari tiga unsur penyusun yaitu:

- **Default Working**, merupakan hari kerja default.
- **Edited Working**, merupakan hari kerja di luar default
- **Edited Non Working**, merupakan hari libur non default



Gambar 1. Tampilan Visual Gantt Chart

Pelaksanaan proyek disusun dalam *task-task*, yang merupakan item *Work Breakdown Structure* (WBS) yang berhasil disusun oleh manajer proyek. WBS menggambarkan struktur hirarki *task* yang mempunyai *outline* berupa *tree*. Pada *Task Sheet*, *task-task* disusun dengan struktur *tree* tersebut. Dalam *Diagram Pane*, *outline* tersebut digambarkan dalam bentuk-bentuk *task* dan *summary task*. Seperti yang ditunjukkan pada gambar 1.

### Metode Diagram Presedensi

*Precedence Diagram Method* (PDM) adalah sebuah teknik untuk menunjukkan hubungan antar aktivitas dengan menggunakan relasi presedensi untuk menunjukkan ketergantungan antar aktivitas.

PDM dapat diterapkan dalam *Gantt Chart* dengan menunjukkan relasi antar *task* yang disebut sebagai *Task Relation* yang secara visual digambarkan sebagai anak panah satu arah. Sebuah *Task Relation* menghubungkan dua *task*, satu adalah *task* yang memberikan batasan yang disebut sebagai *predecessor*, dan lainnya adalah *task* yang diberikan batasan yang disebut sebagai *successor*.

Ada empat jenis *Task Relation* yaitu :

- **Finish To Start (FS)**, menunjukkan bahwa *successor* baru boleh melakukan start setelah *predecessor* melakukan finish.
- **Finish To Finish (FF)**, menunjukkan bahwa *successor* baru boleh melakukan finish setelah *predecessor* melakukan finish.
- **Start To Finish (SF)**, menunjukkan bahwa *successor* baru boleh melakukan finish setelah *predecessor* melakukan start.
- **Start To Start (SS)**, menunjukkan bahwa *successor* baru boleh melakukan start setelah *predecessor* melakukan start.

Masing – masing jenis *Task Relation* tersebut dapat mempunyai waktu tenggang yang disebut sebagai *lag time*. *Lag* dapat mempunyai nilai positif maupun negatif. Contohnya : simbol FS+5 mempunyai arti bahwa *successor* baru boleh melakukan start lima hari setelah *predecessor* melakukan finish.

### Task Constraint

Dalam melakukan penjadwalan berdasarkan presedensi, dibatasi oleh *constraint* yang dimiliki oleh *task* yang diakibatkan oleh faktor teknis maupun non teknis dari pelaksanaan *task* tersebut. *Task Constraint* dibagi menjadi dua yaitu : **flexible constraint** dan **inflexible constraint**. Disebut **flexible constraint** karena tidak tertaut pada batasan tanggal tertentu, sedangkan pada **inflexible** dibatasi tanggal tertentu.

Yang termasuk *Flexible Constraint* adalah :

- **As Soon As Possible (ASAP)**, *task* tersebut dikehendaki untuk dimulai secepat mungkin yang bisa dilakukan.
- **As Late As Possible (ALAP)**, *task* tersebut dikehendaki untuk dimulai seakhir mungkin yang bisa dilakukan terhadap waktu finish proyek yang telah ditentukan.

Yang termasuk *Inflexible Constraint* adalah :

- **Finish No Earlier Than (FNET)**, *task* tersebut tidak boleh finish lebih awal dari tanggal yang telah ditetapkan.
- **Finish No Later Than (FNLTL)**, *task* tersebut tidak boleh finish melebihi dari tanggal yang telah ditetapkan.
- **Must Finish On (MFO)**, *task* tersebut harus finish pada tanggal yang ditetapkan.
- **Must Start On (MSO)**, *task* tersebut harus start pada tanggal yang ditetapkan.
- **Start No Earlier Than (SNET)**, *task* tersebut tidak boleh start lebih awal dari tanggal yang telah ditetapkan.
- **Start No Later Than (SNLTL)**, *task* tersebut tidak boleh finish melebihi dari tanggal yang telah ditetapkan.

### Resource

Yang dimaksud dengan resource dalam konteks *project management* disini adalah unit-unit berupa manusia, perlengkapan maupun material yang diperlukan untuk menyelesaikan *task*.

### Baseline

Baseline adalah sebuah state dari jadwal proyek pada suatu waktu yang sengaja disimpan untuk melakukan *tracking* dan analisa kemajuan proyek. Analisa dilakukan dengan cara membandingkan *task* yang sedang berlangsung dengan baseline *task* tersebut. Baseline dapat dibuat beberapa kali selama proyek berlangsung.

**Java To Enterprise Edition ( J2EE )**

J2EE adalah sebuah platform untuk membangun sebuah aplikasi pada sisi server yang membutuhkan transaksi dalam skala enterprise. Ia merupakan kolaborasi dari beragam konsep, standar pemrograman, dan berbagai inovasi yang semuanya ditulis dalam bahasa pemrograman Java. Kemunculan J2EE ditujukan bagi para pengembang software agar dapat membangun secara cepat aplikasi server-side yang terdistribusi, *scalable*, *reliable*, *portable* dan memiliki sekuriti yang kuat.

J2EE menyediakan suatu lingkungan yang kuat dan *reliable* untuk mengembangkan lapisan *middle-tier*. Adapun beberapa teknologi yang digunakan pada platform J2EE, diantaranya adalah [5]:

- **Enterprise Java Bean (EJB)**  
Enterprise Java Bean mendefinisikan standar dalam penulisan komponen *server-side*, dan menyediakan standar *contract* antara komponen dan *application server* yang mengaturnya.
- **Java Remote Method Invocation (RMI) dan RMI – IIOP**  
RMI menyediakan jalan bagi komunikasi antar proses, dan menyediakan beberapa servis yang berkaitan dengan komunikasi. Sedangkan RMI-IIOP adalah perkembangan dari RMI yang menggunakan protokol Internet Inter-ORB (IIOP) yang dapat diintegrasikan dengan CORBA.
- **Java Naming and Directory Interface ( JNDI )**  
JNDI menyediakan sebuah cara untuk mengidentifikasi lokasi sebuah komponen atau *resource* yang lain dalam sebuah jaringan.
- **Java Database Conectivity ( JDBC )**  
JDBC menyediakan jembatan untuk melakukan akses terhadap database.
- **Java Transaction API ( JTA ) dan Java Transaction Service ( JTS )**  
JTA dan JTS menyediakan beberapa servis transaksi bagi komponen untuk melakukan transaksi.
- **Java Messaging Service ( JMS )**  
JMS menyediakan servis untuk dapat melakukan komunikasi yang bersifat *asynchronous*, dengan sistim komunikasi berupa pesan.
- **Java Servlets dan Java Server Pages (JSPs)**  
Servlet dan JSP ditujukan sebagai kepanjangan dari kemampuan server yang menjadi *host* dari aplikasi yang menggunakan model akses *request-response*. Meskipun sebenarnya dapat melayani *request* dari berbagai tipe, umumnya kebanyakan digunakan pada aplikasi web server.
- **Java IDL**  
IDL adalah implementasi dari CORBA yang dirilis oleh Sun Microsystems. IDL ditujukan untuk menyediakan kemampuan integrasi komponen Java dengan bahasa pemrograman lain.
- **Java Mail**

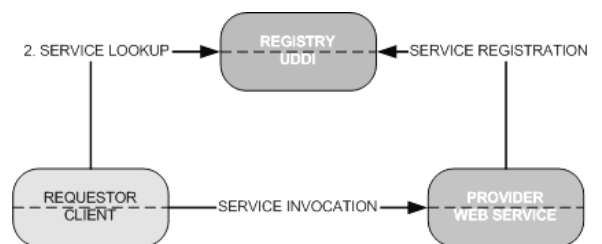
Java Mail ditujukan untuk menyediakan servis e-mail, dengan menggunakan Java.

- **Connectors**  
Connectors ditujukan untuk memberikan kemampuan bagi J2EE agar dapat terintegrasi dengan sistem *mainframe*, yang mempunyai kemampuan melakukan transaksi *high-end*.
- **Extensible Markup Language ( XML )**  
XML adalah sebuah *mark-up language* berbasis text yang ditujukan untuk standarisasi format data.

**Web Service**

Dalam pengertian yang sederhana, *Web Service* adalah cara standar yang dilakukan oleh aplikasi untuk dapat berkomunikasi satu dengan yang lain dengan menggunakan protokol internet standar. Pada lingkungan komputasi terdistribusi, aplikasi didesain sebagai servis yang berjalan pada server. Aplikasi *client* dapat mengakses servis ini melalui sebuah *programmable interface*. Konsep komputasi terdistribusi sebenarnya sudah lama ada, tetapi pendekatan yang dilakukan kebanyakan terikat pada suatu teknologi tertentu (*tightly coupled*). Sedangkan konsep *Web Service* adalah (*loosely coupled*) dalam arti ia bisa dibuat dengan menggunakan berbagai bahasa pemrograman dan dapat berjalan pada berbagai platform, hal ini dikarenakan *Web Service* berkomunikasi dengan menggunakan XML.

Jika melihat dari unsur yang terlibat dalam *Web Service*, dapat dipandang bahwa *Web Service* adalah sebuah jalan komunikasi yang melibatkan tiga peserta yaitu : *provider*, *requestor*, dan *directory*. Hal ini dapat digambarkan seperti pada gambar 2. Dengan menggunakan UDDI (Universal Description, Discovery, and Integration) *requestor* menanyakan kepada *registry* tentang servis yang dibutuhkannya. *Registry* mengembalikan servis – servis yang relevan dengan yang diminta, dan *requestor* memilih satu yang ingin diaksesnya.



**Gambar 2. Hubungan di dalam Web Service.**

**3. PERANCANGAN DAN IMPLEMENTASI SISTEM**

Aplikasi yang akan dibangun merupakan sebuah perangkat lunak visual untuk melakukan manajemen proyek dalam bentuk *Precedence Diagram* yang diterapkan dalam *Gantt Chart*. Dengan antar muka berwujud diagram ini, seorang manajer proyek dapat

membuat jadwal pelaksanaan pekerjaan, melakukan alokasi sumber daya yang dimiliki ke dalam pekerjaan, serta memberikan batasan-batasan pekerjaan yang berupa saling hubungan antar pekerjaan tersebut. Aplikasi ini bersifat *online* menggunakan jaringan internet sebagai media untuk mendukung kemampuan *multiuser*, sehingga dapat digunakan bersama oleh beberapa user sesuai dengan otoritas yang telah diberikan.

Pada aplikasi *Gantt Chart* online yang dibuat pada penelitian ini, beberapa elemen dasar yang akan dibangun meliputi *Task Sheet*, *Time Line*, *Work Calendar*, dan *Diagram Pane*.

### Task Sheet

Pada sebuah *Gantt Chart*, sebuah *Task Sheet* terletak pada bagian sebelah kiri, berfungsi sebagai sumbu vertikal. Pada kebanyakan aplikasi yang ada, *Task Sheet* diwujudkan sebagai antar muka visual dalam bentuk sebuah grid dengan tiap baris atau *row* merepresentasikan sebuah *task* dan tiap kolomnya merupakan atribut-atribut yang dimiliki oleh *task* tersebut.

### Time Line

Sebuah project mempunyai unsur waktu sebagai *limit* yang harus dipenuhi, karena itu aktivitas-aktivitas dalam proyek harus pula dapat diukur dalam satuan-satuan waktu. Sebagai acuan waktu secara visual, dalam sebuah *Gantt Chart* diberikan *Time Line* berupa rentang waktu dalam sumbu horisontal, terletak pada bagian atas dari *Diagram Pane*.

### Work Calendar

*Work Calendar* digunakan untuk menentukan bagaimana *resource* yang dialokasikan ke *task* dijadwalkan, dan juga bagaimana *task* itu sendiri dijadwalkan. Dalam aplikasi manajemen proyek ini, *Work Calendar* diasosiasikan sebagai hari kerja aktif, yang akan mempengaruhi durasi proyek, berdasarkan kepada sistem kalender yang digunakan.

### Diagram Pane

Dalam *Gantt Chart*, terdapat sebuah panel yang digunakan untuk menggambar diagram visual sebuah proyek yang disebut *Diagram Pane*. Panel ini terletak dibawah serta sejajar dengan *Time Line* karena pada dasarnya *Time Line* digunakan sebagai skala waktu dari *Diagram Pane*. Panel tersebut mempunyai latar belakang berupa garis-garis vertikal yang terhubung ke periode mingguan pada *Time Line*, untuk memudahkan pembacaan waktu terhadap diagram, serta garis-garis blok vertikal yang menunjukkan sebagai hari libur. Letak *Diagram Pane* pada *Gantt Chart* ditunjukkan seperti pada gambar 1.

### Task Constraint

Sebuah *task* dapat mempunyai *constraint* yang membatasi waktu pelaksanaan. *Constraint* ini dibedakan menjadi dua yaitu *flexible constraint* dan *inflexible constraint*. Perbedaannya, *flexible constraint* tidak ditautkan pada tanggal spesifik tertentu, sementara *inflexible constraint* ditautkan secara ketat terhadap suatu tanggal dalam *Time Line*.

#### Flexible Constraint :

##### - *As Soon As Possible (ASAP)*

Secara default, apabila kita menjadwalkan proyek dari waktu *start* proyek, *constraint* yang berlaku pada *task* yang ditambahkan adalah ASAP, yaitu kita menghendaki supaya *task* dapat dimulai secepat mungkin. Apabila pada *task* tidak mempunyai *task relation* terhadap *task* lain yang membatasi, maka *start task* tersebut akan disamakan dengan waktu *start* proyek, karena hal itu adalah *start* tercepat yang dapat dilakukan. *Constraint* ASAP ini tidak ditautkan pada tanggal tertentu.

##### - *As Late As Possible (ALAP)*

Secara default, apabila kita menjadwalkan proyek dari waktu *finish* proyek, *constraint* yang berlaku pada *task* yang ditambahkan adalah ALAP, yaitu kita menghendaki supaya *task* dimulai pada waktu akhir maksimal yang mungkin, dilihat terhadap waktu *finish* proyek. Apabila pada *task* tidak mempunyai *task relation* terhadap *task* lain yang membatasi, maka *start task* tersebut adalah waktu *finish* proyek dikurangi dengan durasi *task*.

#### Inflexible Constraint :

##### - *Finish No Earlier Than (FNET)*

Secara default, pada penjadwalan yang dimulai pada *start* proyek, apabila kita memberikan nilai tanggal tertentu kepada *finish* dari sebuah *task*, maka secara otomatis tanggal tersebut akan menjadi *constraint* FNET, yang berarti *task* tersebut tidak boleh atau tidak dapat diselesaikan lebih awal dibandingkan dengan tanggal tersebut.

##### - *Finish No Later Than (FNLT)*

*Constraint* FNLT memberikan batasan bahwa sebuah *task* tidak diperbolehkan untuk selesai melebihi waktu yang ditentukan, karena misalnya akan menimbulkan kerugian atau bahaya tertentu.

##### - *Must Finish On (MFO)*

*Constraint* MFO terikat sangat ketat dan harus tepat terhadap suatu tanggal tertentu. MFO mempunyai makna bahwa *task* harus selesai tepat pada tanggal tertentu, sehingga konsekuensinya apabila diberikan *task relation* yang akan menggeser waktu *finish* dari *task* tersebut, harus dikeluarkan peringatan atau *alert* karena *task relation* tersebut tidak boleh diberikan.

- **Must Start On (MSO)**  
Constraint MSO bersifat seperti MFO, hanya ia berlaku terhadap *start* sebuah *task*, yaitu *task* tersebut harus tepat dimulai pada suatu tanggal tertentu.
- **Start No Earlier Than (SNET)**  
Secara default, pada penjadwalan yang dimulai pada *start* proyek, apabila kita memberikan nilai tanggal tertentu kepada *start* dari sebuah *task*, maka secara otomatis tanggal tersebut akan menjadi *constraint* SNET, yang berarti *task* tersebut tidak boleh atau tidak dapat dimulai lebih awal dibandingkan dengan tanggal tersebut.
- **Start No Later Than (SNLT)**  
Constraint SNLT memberikan batasan bahwa sebuah *task* tidak diperbolehkan untuk dimulai pada waktu melebihi dari tanggal tertentu yang telah ditentukan, seperti misalnya disebabkan jika melebihi waktu tersebut, maka bahan baku produksi yang akan dibuat akan mulai rusak, dsb.

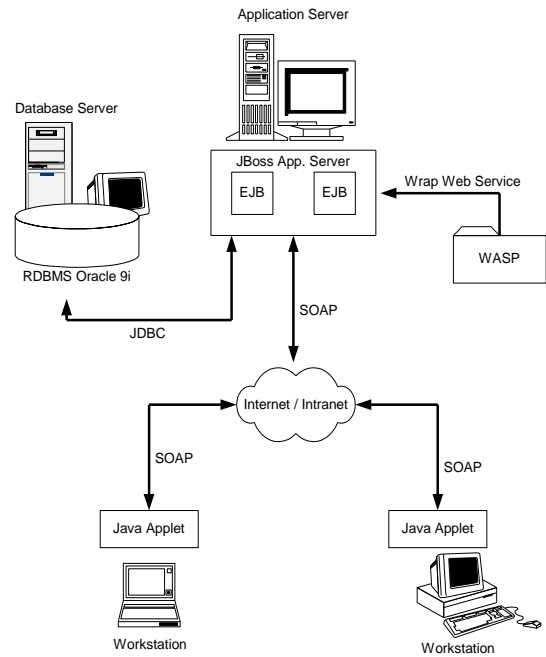
### Task Relation

Dalam *Diagram Presedensi* dikenal ada empat macam hubungan antar *task* atau biasa disebut *task relation*, yaitu :

- **Finish to Start (FS)**  
Relasi *Finish to Start* menunjukkan sebuah batasan bahwa sebuah *task* yang menjadi *successor* tidak boleh melakukan *start* sebelum *task* yang menjadi *predecessor* selesai dikerjakan.
- **Finish to Finish (FF)**  
Relasi *Finish to Finish* menunjukkan batasan bahwa sebuah *task* yang menjadi *successor* tidak boleh diakhiri atau *finish* sebelum *task* yang menjadi *predecessor* selesai dikerjakan.
- **Start to Finish (SF)**  
Pada relasi *Start to Finish*, *task successor* hanya dapat diakhiri jika *task predecessor* siap untuk dimulai.
- **Start to Start (SS)**  
Pada relasi *Start to Start*, pekerjaan atau *task successor* hanya dapat dimulai jika *predecessornya* sudah dimulai dikerjakan.

### Arsitektur Sistem

Berangkat dari spesifikasi kebutuhan bahwa aplikasi yang dibuat harus dapat digunakan secara multiuser, maka perangkat lunak manajemen online ini dibuat berbentuk aplikasi *Client – Server* berbasis web dengan arsitektur *3-tier* yang terdiri atas *user interface layer*, *business logic layer*, dan *database layer*, seperti yang dapat dilihat pada gambar 3.



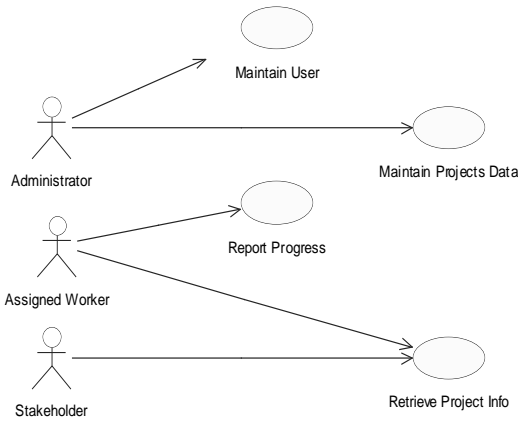
Gambar 3. Arsitektur sistem

### User Actor

Dalam aplikasi manajemen proyek ini, terdapat empat *User Actor* yang berkepentingan dan dapat melakukan interaksi dengan aplikasi. Keempat *actor* tersebut adalah : *Administrator*, *Project Manager*, *Worker*, dan *Project Stakeholder*.

- **Administrator**  
Seorang administrator mempunyai wewenang untuk mengatur manajemen user berdasarkan wewenangnya untuk mengakses aplikasi manajemen proyek, serta melakukan pemeliharaan terhadap data–data proyek, seperti misalnya kegiatan backup data–data proyek, atau menghapus data yang dinilai tidak dibutuhkan lagi. Diagram *Use Case* untuk *actor* Administrator dapat dilihat seperti gambar 4.
- **Project Manager**  
Manajer proyek adalah orang yang membuat perencanaan proyek atau *project plan*. *Use Case Diagram* untuk *actor* Project Manager, dapat dilihat seperti dalam gambar 5.
- **Worker**  
*Worker* adalah *actor* yang dalam prakteknya merupakan pelaksana kegiatan proyek. Jadi sewajarnya ia adalah termasuk anggota *resource* yang dialokasikan untuk menyelesaikan *task*. *Actor* ini berkepentingan untuk melihat spesifikasi *task* dimana ia ditugaskan, serta berkewajiban melaporkan progress yang telah dikerjakannya. *Use Case Diagram* untuk *actor* Worker dapat dilihat seperti pada gambar 4.
- **Project Stakeholder**  
*Stakeholder* adalah *actor* yang berkepentingan untuk meninjau sebuah proyek. Mungkin ia adalah seorang pemegang saham, atau pemberi

proyek. Use case Diagram untuk actor Stakeholder dapat dilihat seperti pada gambar 4.



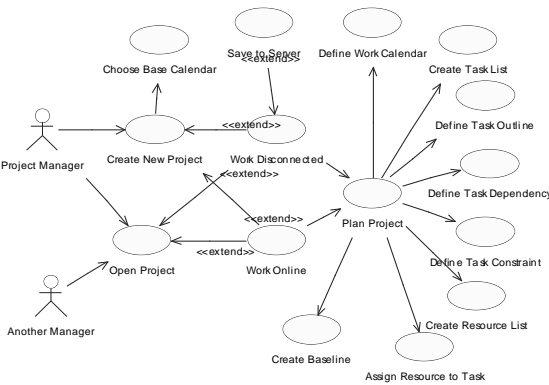
**Gambar 4. Use Case Diagram Actor Administrator, Worker, Stakeholder.**

**4. HASIL UJI COBA**

Perancangan aplikasi yang dibuat telah berhasil diimplementasikan dengan menggunakan teknologi dan framework yang telah ditetapkan. Uji Coba yang telah dilakukan meliputi deployment aplikasi pada sebuah Local Area Network, dan menjalankan beberapa aplikasi client project management dari workstation yang berbeda, kemudian melakukan kolaborasi perencanaan sebuah proyek. Kronologi pembuatan perencanaan project yang dilakukan adalah :

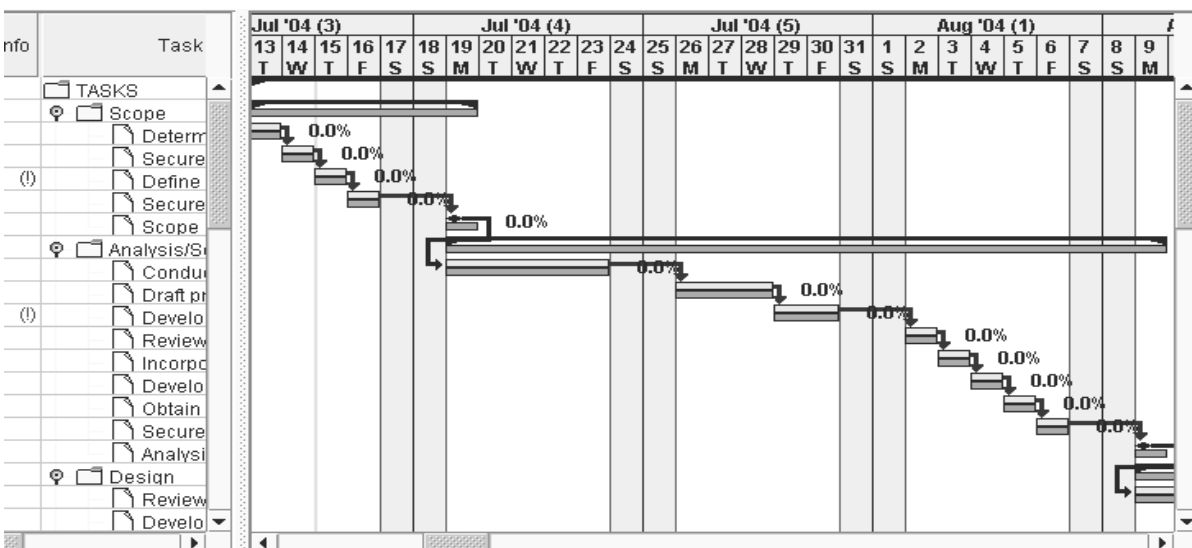
1. Membuat Sebuah Project Baru
2. Beberapa User Bergabung pada Project
3. Merencanakan Work Calendar
4. Membuat Task List
5. Menentukan Outline dari Task List
6. Merencanakan Task Constraint
7. Merencanakan Task Deadline
8. Membuat dependensi task dengan menggunakan Task Relation
9. Merencanakan Resource project
10. Membuat baseline untuk tracking

Dari kronologi uji coba yang dilakukan tersebut, telah berhasil didapatkan kesesuaian diagram antar user dengan pendekatan sifat real time. Gambar 6 menunjukkan sebuah snapshot dari hasil uji coba yang dilakukan.



**Gambar 5. Use Case Diagram Actor Project Manager**

Aplikasi Project Management sudah tersedia cukup banyak di pasar perangkat lunak, contohnya yang cukup dikenal adalah : Microsoft Project dan Primavera. Sebagai perbandingan, dalam Tabel 1 disajikan perbandingan fitur – fitur yang disediakan dalam aplikasi Project Management yang dibuat dalam penelitian ini dengan Microsoft Project Professional 2002 dan Primavera Versi 3.1.



**Gambar 6. Hasil Uji Coba Aplikasi**





Tabel 1. Tabel Perbandingan Fitur Aplikasi

No	FITUR DISEDIAKAN	MP	PRM	PL
1	Gantt Chart View	√	√	√
2	Network Diagram View	√	√	X
3	Tracking Gantt	√	√	√
4	Resource Usage View	√	√	X
5	Cost Calculation	√	√	X
6	Multi Project Environment	√	√	X
7	Calendar Per Resource	√	√	X
8	Online Collaboration	√	√	√
9	Realtime Online Collaboration Editing	X	X	√
10	User Management System	√	√	√
11	Multi Operating System Platform	X	X	√

Keterangan :

- √ = tersedia
- X = tidak tersedia
- MP = Microsoft Project 2002
- PRM = Primavera 3.1
- PL = Aplikasi Project Management dalam Penelitian ini

Tampak bahwa keunggulan dari aplikasi ini adalah pemanfaatannya yang dapat digunakan bersama-sama secara online serta tidak terbatas pada sistem operasi tertentu.

## 5. KESIMPULAN

Berdasarkan pada hasil perancangan dan pembuatan perangkat lunak aplikasi *project management* online, terhadap permasalahan yang dirumuskan, dapat diambil beberapa kesimpulan sebagai berikut :

1. Untuk membuat aplikasi terdistribusi, aspek portabilitas terhadap sistem operasi sangat penting untuk diperhatikan, Java adalah sebuah teknologi yang ditujukan untuk hal tersebut. Aplikasi *Project Management*, khususnya komponen *Gantt Chart* dalam penelitian ini telah berhasil dibuat dengan menggunakan Java, sehingga menjadi bersifat portabel.
2. Metode Diagram Presedensi dapat diimplementasikan dalam Aplikasi *Project Management*, metode ini dapat membantu dalam melakukan penjadwalan *task project*.

3. Dengan menerapkan teknologi aplikasi terdistribusi menggunakan *framework* J2EE dalam hal ini menggunakan JBoss *application server* sebagai *EJB Container*, telah berhasil dibuat aplikasi *Project Management* yang bersifat *online* dan *multiuser*. Pendekatan *real time* dilakukan dengan sistem *update polling* aplikasi *client* terhadap server.
4. Sebagai aplikasi *online*, idealnya aplikasi *Project Management* dioptimalkan untuk menghadapi resiko restriksi internet seminimal mungkin. Aplikasi ini telah berhasil menerapkan sistem komunikasi berbasis *Web Service* menggunakan *WASP Server*, sehingga tingkat restriksinya sama dengan yang dimiliki oleh web.
5. Aplikasi ini telah berhasil disusun berdasarkan konsep arsitektur aplikasi *three-tier* sehingga keuntungan konsep ini bisa didapat, seperti biaya *deployment* yang dapat direduksi, *maintenance* perubahan *business logic* menjadi mudah, dapat dilakukan *resource pooling* komponen server dll.
6. Sebagai aplikasi *case tool* visual, aplikasi ini telah berhasil dibuat interaktif terhadap user dengan menggunakan *library* GUI yang disediakan oleh java yang menyediakan fitur-fitur yang memadai.

## 6. DAFTAR PUSTAKA

1. Harold Kerzner PhD, *Project Management A System Approach to Planning, Scheduling and Controlling*, John Wiley & Sons Inc., 2001.
2. Ilman Soeharto, *Manajemen Proyek Dari Konseptual Sampai Operasional*, Erlangga, Jakarta, 1997.
3. Clemen Szyperski, *Component Software Beyond Object Oriented Programming*, Addison Wesley, 1999.
4. Sun Microsystems, *Java Web Service Tutorial*, 2002.
5. Ed Roman, *Mastering Enterprise Java Beans and The Java 2 Platform, Enterprise Edition*, John Wiley & Sons Inc., 1999.
6. Kevin Williams, *XML Structures for Existing Database*, [www-136.ibm.com/developerworks/xml](http://www-136.ibm.com/developerworks/xml), 2001.
7. Systinet Corp., *Developers Corner :Web Services & J2EE*, 2003.
8. The Numbers Group, *Project Management*, [www.snc.edu/socsci/chair](http://www.snc.edu/socsci/chair)