



**PENGEMBANGAN SISTEM INFORMASI PENJUALAN  
PROPERTI BERBASIS WEB MENGGUNAKAN METODE  
RAPID APPLICATION DEVELOPMENT (RAD)  
(STUDI KASUS : PERUSAHAAN MP PROJECT)**

**TUGAS AKHIR**

PUTUT BUDIUTOMO  
2020081018

PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNOLOGI DAN DESAIN  
UNIVERSITAS PEMBANGUNAN JAYA  
TANGERANG SELATAN  
2023/2024



## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>i</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Identifikasi Masalah.....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat Penelitian .....	3
1.5 Sistematika Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>7</b>
2.1 Teori Dasar.....	7
2.1.1 Pengertian Sistem .....	7
2.1.2 Pengertian Informasi.....	8
2.1.3 Pengertian Sistem Informasi .....	8
2.1.4 Pengertian Sistem Informasi Pemasaran.....	9
2.1.5 Pengertian Pengembangan Sistem Informasi .....	9
2.2 Teori Khusus .....	10
2.2.1 Metode Rapid Application Development .....	10
2.3 Tinjauan Studi.....	12
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>17</b>
3.1 Objek Penelitian .....	17
3.1.1. Metode Penelitian .....	18

3.1.2. Tahap Pengumpulan Data .....	19
3.1.3. Tahap Pengembangan Sisten .....	19
<b>BAB 4 BIAYA DAN JADWAL KEGIATAN.....</b>	<b>21</b>
4.1 Jadwal Implementasi.....	21
4.2 Rencana Anggaran Biaya.....	21
<b>DAFTAR PUSTAKA.....</b>	<b>23</b>
<b>LAMPIRAN-LAMPIRAN .....</b>	<b>25</b>
Lampiran 1. Justifikasi Anggaran Kegiatan .....	25

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi informasi telah memberikan dampak yang signifikan pada berbagai aspek kehidupan, termasuk dalam dunia bisnis dan industri. Industri properti turut mengadopsi kemajuan teknologi ini. MP Project, sebagai pengembang perumahan, sedang fokus dalam upaya penjualan properti. Mereka menyadari pentingnya memanfaatkan teknologi untuk mendukung proses penjualan dan memberikan pengalaman lebih baik kepada calon konsumen. Proses penjualan properti oleh MP Project dimulai dari promosi dan pemasaran, diikuti dengan kunjungan calon konsumen ke lokasi properti. Selama kunjungan, tim penjualan memberikan informasi detail tentang properti dan menjawab pertanyaan calon konsumen. Tahap selanjutnya adalah negosiasi harga dan ketentuan pembayaran, di mana MP Project berupaya mencapai kesepakatan yang menguntungkan. Setelah kesepakatan, calon konsumen melakukan pembayaran sesuai kesepakatan dan menandatangani kontrak pembelian properti.

Namun, dalam beberapa proses penjualan, MP Project menghadapi kendala. Kendala pertama terkait dengan proses pemasaran properti yang masih mengandalkan brosur fisik. Brosur ini memiliki keterbatasan dalam memberikan informasi lengkap dan visual yang memadai tentang properti, serta sulit diperbarui sesuai informasi terkini. Selanjutnya, dalam manajemen properti, MP Project tidak memiliki sistem terintegrasi yang memungkinkan pengelolaan informasi properti. Informasi terperinci seperti detail, spesifikasi, foto, lokasi, dan fasilitas tersebar di berbagai platform *online* yang tidak terhubung dengan baik. Ketidakterhubungan ini menyulitkan penyajian data yang konsisten dan akurat kepada tim penjualan dan calon konsumen.

Kendala berikutnya yang dihadapi oleh MP Project terkait dengan pemberkasan dokumen konsumen yang proses pengorganisasian dan pengelolaan yang kurang teratur terhadap berkas-berkas tersebut. Dokumen-dokumen seperti foto kopi Kartu Tanda Penduduk (KTP), Nomor Pokok Wajib Pajak (NPWP), Kartu Keluarga (KK), dan dokumen-dokumen lain yang berhubungan dengan persyaratan pembayaran seperti *cash* keras, *cash* bertahap, atau Kredit Pemilikan Rumah (KPR) menjadi bagian penting dari proses verifikasi dan administrasi dalam penjualan properti. Masalah utama yang dihadapi adalah ketidakteraturan dalam penyimpanan, pelabelan, dan pengelompokan berkas-berkas ini. Dokumen-dokumen tersebut sering kali tersebar di berbagai lokasi atau format, sehingga sulit untuk diakses dan dikelola dengan efisien. Selain itu, proses verifikasi dan pengecekan dokumen-dokumen ini bisa memakan waktu yang lama karena kesulitan dalam melacak status, keberadaan, dan kelengkapannya. Kurangnya pengorganisasian dalam pengelolaan dokumen konsumen menyebabkan potensi kesalahan, kehilangan dokumen, atau keterlambatan dalam proses verifikasi. Hal ini tidak hanya mempengaruhi kecepatan dalam menyelesaikan transaksi penjualan properti, tetapi juga dapat mengganggu pengalaman calon konsumen.

Untuk mengatasi kendala-kendala tersebut, peneliti mengusulkan implementasi sistem yang mampu menyajikan informasi properti secara komprehensif melalui pengelolaan data properti yang terpusat. Sistem ini akan memungkinkan pengintegrasian seluruh informasi properti ke dalam satu platform terpadu. Dengan demikian, tim penjualan dan calon konsumen dapat mengakses informasi properti yang lengkap, terkini, serta konsisten dari satu sumber yang terpercaya. Selain itu, peneliti juga menyarankan sistem yang mampu mengorganisasikan dokumen-dokumen konsumen secara terpusat dan terstruktur. Implementasi sistem ini akan memungkinkan penyimpanan, pelabelan, dan manajemen dokumen-dokumen terkait konsumen ke dalam satu tempat yang terpadu. Dengan adanya sistem ini, akan lebih mudah untuk melakukan pelacakan status

dokumen, memastikan kelengkapan, serta mempercepat proses verifikasi yang diperlukan dalam transaksi penjualan properti.

Dari pertimbangan tersebut, solusi yang diusulkan adalah pengembangan sistem informasi penjualan properti berbasis web dengan metode Rapid Application Development (RAD). Dengan sistem ini, diharapkan MP Project dapat mengatasi berbagai kendala yang diidentifikasi, terutama dalam validasi data konsumen, pengelolaan data penjualan yang terintegrasi, serta dalam proses administrasi penjualan. Oleh karena itu, judul penelitian ini adalah **"PENGEMBANGAN SISTEM INFORMASI PENJUALAN PROPERTI BERBASIS WEB DENGAN METODE RAPID APPLICATION DEVELOPMENT (RAD) STUDI KASUS : PERUSAHAAN MP PROJECT"**.

## **1.2 Identifikasi Masalah**

Melalui tinjauan mendalam terhadap proses penjualan, masalah-masalah muncul dengan jelas. Tantangan-tantangan ini menjadi fokus utama dalam penelitian ini, yang bertujuan untuk menghasilkan solusi inovatif dalam bentuk sistem informasi penjualan properti berbasis web. Dalam rangkaian ini, berikut adalah identifikasi mengenai masalah-masalah yang dihadapi oleh MP Project di antaranya adalah :

- a. Informasi terperinci tentang properti, seperti file foto, dokumen-dokumen spesifikasi bangunan tersebar di berbagai platform *online* yang tidak terintegrasi dengan baik. Hal ini menyulitkan penyajian data yang konsisten dan akurat kepada tim penjualan dan calon konsumen.
- b. Sistem pengelolaan properti yang ada tidak mampu mengelola informasi properti secara *real-time*, mengakibatkan kesulitan dalam pembaruan dan akses informasi terkini, serta menghambat respons cepat terhadap perubahan dan kebutuhan pasar.
- c. Dokumen-dokumen penting seperti foto kopi KTP, NPWP, KK, dan lainnya terkait persyaratan pembayaran tidak terorganisir

dengan baik. Hal ini mengakibatkan kesulitan dalam pelacakan status, kelengkapan, dan proses verifikasi dokumen, yang dapat memperlambat transaksi penjualan properti.

### **1.3 Rumusan Masalah**

Berdasarkan identifikasi masalah yang telah diuraikan, dapat disimpulkan mengenai rumusan masalah yang sesuai adalah “Bagaimana mengembangkan sebuah sistem informasi penjualan properti pada perusahaan MP Project dengan menggunakan metodologi pengembangan *Rapid Application Development* (RAD) ?

### **1.4 Tujuan Penelitian**

Dengan menjawab pertanyaan tersebut, peneliti dapat mengidentifikasi solusi yang tepat dalam merancang dan membangun sistem informasi pemasaran properti berbasis web yang dapat mengatasi kendala yang dihadapi oleh perusahaan MP Project dalam penjualan properti.

Adapun tujuan dari penelitian ini sebagai berikut :

1. Mengembangkan sistem yang mampu mengelola informasi properti secara *real-time*, memungkinkan pembaruan dan akses informasi terkini untuk respons yang cepat terhadap perubahan pasar dengan penyajian data properti yang komprehensif, akurat, dan terkini.
2. Merancang sistem yang dapat mengorganisasikan dokumen-dokumen konsumen secara terstruktur dan terpusat, memungkinkan pelacakan status, memastikan kelengkapan, dan mempercepat proses verifikasi dokumen untuk transaksi penjualan properti.
3. Mengimplementasikan sistem yang dapat mengatasi kendala yang dihadapi, sehingga meminimalisir potensi kesalahan, keterlambatan, atau kehilangan dokumen dalam proses transaksi penjualan properti MP Project.



Dengan mencapai tujuan-tujuan ini, diharapkan dapat membantu MP Project meningkatkan kinerja bisnis penjualan properti mereka dan memberikan pengalaman yang lebih baik kepada calon konsumen.

### **1.5 Manfaat Penelitian**

Penelitian ini diharapkan akan memberikan berbagai manfaat, baik bagi MP Project sebagai perusahaan pengembang properti maupun bagi perkembangan teknologi informasi dalam industri properti. Adapun manfaat dari penelitian ini adalah:

1. Memberikan MP Project sistem yang mampu mengelola informasi properti secara *real-time*, mempercepat respons terhadap perubahan pasar, dan meningkatkan akses terhadap informasi terkini.
2. Memungkinkan penyajian data properti yang komprehensif, akurat, dan terkini kepada calon konsumen, yang dapat meningkatkan pengalaman mereka dalam proses penjualan properti.
3. Dengan pengelolaan dokumen konsumen yang terorganisir, mempercepat proses verifikasi dokumen dan transaksi penjualan properti dan mengurangi potensi kesalahan, keterlambatan, atau kehilangan dokumen dalam proses transaksi.

Dengan demikian, hasil dari penelitian ini tidak hanya memberikan manfaat praktis kepada perusahaan MP Project, tetapi juga memberikan kontribusi berharga dalam perkembangan ilmu pengetahuan, teknologi, dan dapat dimanfaatkan oleh peneliti lain dan masyarakat luas.

### **1.6 Sistematika Penulisan**

Adapun sistematika yang digunakan penulis dalam menyusun penelitian untuk memudahkan pemahaman yang lebih jelas mengenai struktur penulisan. Berikut adalah sistematika penulisan dalam penelitian ini :

a. BAB I PENDAHULUAN

Bab ini memperkenalkan latar belakang masalah yang melatarbelakangi penelitian, mengidentifikasi masalah yang dihadapi oleh perusahaan MP Project dalam penjualan dan pengelolaan properti. Di sini, juga diuraikan rumusan masalah, tujuan penelitian, manfaat penelitian, dan gambaran singkat mengenai struktur keseluruhan penulisan.

b. BAB II: TINJAUAN PUSTAKA

Bagian ini menjelaskan mengenai beberapa teori dan penjelasan ilmu yang berhubungan dengan penelitian. Teori-teori tersebut dijelaskan secara sistematis dengan memuat fakta, serta hasil penelitian sebelumnya, yang berasal dari pustaka.

c. BAB III METODE PENELITIAN

Pada bab ini, akan dijelaskan metode penelitian yang digunakan. Penelitian ini dilakukan dengan mengumpulkan data melalui wawancara dengan pihak terkait, observasi langsung terhadap kegiatan operasional yang berjalan, dan studi dokumentasi. Dengan menggunakan metode ini, diharapkan sistem yang dibangun dapat memenuhi kebutuhan yang ada dan memberikan manfaat yang maksimal bagi perusahaan MP Project.

d. BAB IV HASIL DAN ANALISIS PENELITIAN

Dalam bab ini, hasil dari penelitian akan dipresentasikan dengan jelas. Temuan-temuan yang ditemukan dalam pengembangan sistem informasi pemasaran properti berbasis web akan dijabarkan secara detail dengan rancangan sistem menggunakan metode UML (Unified Modelling Language), perancangan tampilan, serta implementasi.

e. BAB V PENUTUP

Bab terakhir ini akan menguraikan kesimpulan yang diambil dari hasil penelitian. Dari temuan-temuan yang telah dianalisis,

akan dijabarkan implikasi praktis dan kontribusi penelitian ini terhadap perkembangan ilmu pengetahuan. Selain itu, rekomendasi dan saran untuk pengembangan lebih lanjut juga akan dijelaskan di bab ini.

Dengan mengikuti sistematika penulisan yang terstruktur ini, diharapkan pembaca dapat mengikuti alur penelitian dengan baik dan memperoleh pemahaman yang komprehensif mengenai masalah, metodologi, temuan, serta kontribusi penelitian ini.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

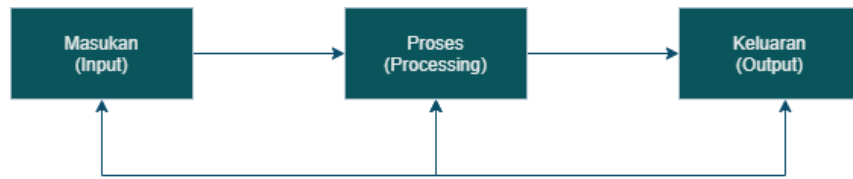
##### **2.1.1 Pengertian Sistem**

Pembahasan mengenai metode dalam pengembangan sistem tidak dapat dipisahkan dari konsep sistem informasi. Memahami sistem informasi menjadi hal yang sangat penting sebagai langkah awal dalam proses pengembangan sistem.

Sistem adalah kumpulan komponen atau elemen-elemen yang saling terhubung dan saling memengaruhi satu sama lain, bekerja bersama secara harmonis, dan memiliki tujuan tertentu. Sistem dapat berbentuk konsep abstrak atau benda fisik, dan dalam sistem terdapat komponen-komponen seperti masukan (input), proses (pengolahan), dan keluaran (output). Tujuan utama dari sistem adalah mencapai suatu tujuan khusus melalui interaksi yang teratur dan terkoordinasi antara komponen-komponen yang ada dalam sistem. Di bawah ini merupakan pengertian sistem menurut berbagai pakar :

(Permana , Nur, & Rahma, 2022) menjelaskan sistem dapat juga diartikan sebagai himpunan elemen yang saling digabungkan untuk mencapai tujuan bersama. Sistem bisa berupa konsep abstrak atau benda nyata. Sistem merupakan kelompok komponen, baik yang berwujud fisik maupun yang tidak fisik, yang saling terhubung dan berfungsi bersama secara serasi untuk mencapai suatu tujuan khusus.

(Prabowo, 2020) mendefinisikan sistem sebagai kumpulan komponen atau elemen-elemen yang saling terhubung dan memengaruhi satu sama lain dalam melakukan kerja sama untuk mencapai suatu tujuan tertentu. Umumnya, komponen-komponen yang ada dalam sistem meliputi *input* (masukan), proses (pengolahan), dan *output* (keluaran). Berikut merupakan gambar dari model komponen sistem :



*Gambar 2.1. Model Sistem (Prabowo, 2020, p. 2)*

Pada Gambar 1, dijelaskan bahwa setiap sistem minimal harus terdiri dari tiga komponen utama, yaitu elemen masukan, proses pengelolaan, dan komponen keluaran. Selain aspek tersebut, penting juga untuk memperhatikan adanya mekanisme umpan balik atau kontrol dalam sistem tersebut.

### **2.1.2 Pengertian Informasi**

Menurut (Hidayat, 2019) Informasi adalah hasil dari pengolahan data yang memberikan nilai dan kegunaan bagi organisasi. Data dalam bentuk mentahnya tidak memiliki nilai yang signifikan tanpa melalui proses transformasi dan analisis yang tepat. Namun, ketika data diolah dengan menggunakan metode atau algoritma yang sesuai, informasi yang bernilai dapat dihasilkan.

Informasi dapat memberikan pemahaman yang lebih mendalam tentang situasi, tren, pola, atau kondisi yang ada di dalam organisasi. Oleh karena itu, informasi menjadi aset penting bagi organisasi karena dapat digunakan sebagai dasar pengambilan keputusan yang cerdas dan efektif, memungkinkan organisasi untuk mengoptimalkan operasional, meningkatkan efisiensi, dan mencapai tujuan bisnis yang ditetapkan.

### **2.1.3 Pengertian Sistem Informasi**

(Prehanto, S.Kom., M.Kom, 2020) menjelaskan Sistem informasi merupakan suatu proses yang melibatkan pengumpulan, penyimpanan, dan analisis informasi untuk mencapai tujuan tertentu. Sistem informasi terdiri dari data masukan (input) dan menghasilkan keluaran berupa laporan (output) yang kemudian diterima oleh sistem lain. Sistem ini juga mendukung kegiatan strategis di dalam sebuah organisasi untuk memandu tindakan dan pengambilan keputusan.

Dalam sebuah organisasi, sistem informasi dirancang dan diterapkan untuk mendukung berbagai kegiatan dan proses bisnis yang ada. Tujuannya adalah agar kegiatan organisasi dapat berjalan dengan lancar, efisien, dan efektif, serta dapat mencapai tujuan yang telah ditetapkan. Sistem informasi berfungsi sebagai pendukung utama dalam mengumpulkan, mengelola, dan menganalisis data, yang kemudian diubah menjadi informasi yang bernilai.

(Permana , Nur, & Rahma, 2022) mengatakan bahwa saat ini, fokus utama dalam pengembangan sistem informasi adalah pada sistem informasi yang berbasis komputer. Tujuannya adalah untuk meningkatkan kecepatan, keakuratan, dan kualitas pengolahan data, dengan harapan dapat mencapai pengambilan keputusan yang lebih efisien dan efektif. Meskipun demikian, sistem informasi berbasis komputer tidak selalu mengarah pada otomatisasi total. Hal ini dikarenakan ada aktivitas tertentu yang lebih baik dilakukan oleh manusia, sementara yang lain dapat dilakukan oleh mesin. Dalam praktiknya, dikembangkan sistem hybrid yang menggabungkan interaksi manusia dan mesin untuk mencapai hasil terbaik.

Dengan demikian, sistem informasi merupakan suatu komponen penting dalam keberhasilan dan pertumbuhan sebuah organisasi, karena membantu dalam pengolahan data dan informasi yang mendukung kegiatan bisnis serta mencapai tujuan yang telah ditetapkan.

#### **2.1.4 Pengertian Penjualan**

Penjualan adalah proses yang dilakukan oleh seorang penjual dengan tujuan untuk memperoleh keuntungan dari penjualan barang atau jasa. Hal ini melibatkan transfer hak kepemilikan atas barang atau jasa dari penjual ke pembeli. Terdapat sejumlah persyaratan yang harus dipenuhi agar penjualan dapat terjadi, termasuk adanya penjual dan calon pembeli, terjadinya interaksi dan persepsi, niat untuk melakukan pertukaran atau transaksi, serta adanya barang, jasa, ide, gagasan, rencana, keyakinan, dan prinsip yang diperdagangkan (Mulyadi, 2008).

Kegiatan penjualan adalah salah satu aspek yang sangat vital dan sering kali terjadi dalam operasional setiap perusahaan dagang. (Mulyadi, 2009) menjelaskan Kegiatan penjualan dapat dilakukan oleh perusahaan dalam beberapa bentuk yaitu :

1. Penjualan Tunai

Penjualan tunai dilakukan dengan cara mewajibkan pembeli untuk membayar harga barang terlebih dahulu sebelum barang diserahkan kepada mereka. Setelah perusahaan menerima pembayaran, barang kemudian diserahkan kepada pembeli, dan transaksi penjualan tunai dicatat.

2. Penjualan Cicilan

Penjualan cicilan adalah penjualan barang atau jasa yang dilakukan melalui perjanjian di mana pembayaran dilakukan secara bertahap atau angsuran. Biasanya, saat barang atau jasa diserahkan kepada pembeli, penjual menerima pembayaran pertama (uang muka) dan sisanya dibayar dalam beberapa angsuran. Karena penjual harus menunggu beberapa periode untuk menerima seluruh pembayaran, mereka mungkin menetapkan bunga atas saldo yang masih belum dibayarkan.

### **2.1.5 Pengembangan Sistem Informasi**

(Handayani , et al., 2023) menjelaskan bahwa pengembangan sistem informasi dapat didefinisikan sebagai proses kegiatan yang bertujuan untuk menciptakan sistem informasi berbasis komputer guna menyelesaikan permasalahan yang ada dalam sebuah organisasi atau untuk mengambil keuntungan dari peluang-peluang yang muncul.

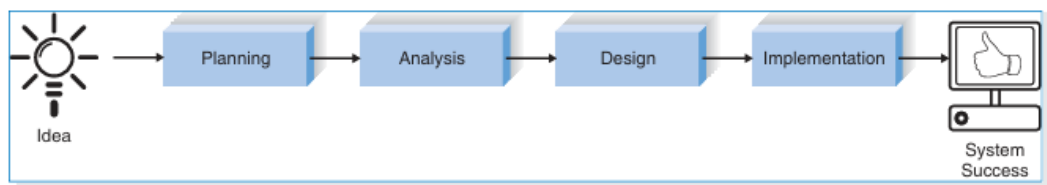
Terdapat beberapa metode dalam mengembangkan sistem informasi, dan yang paling umum digunakan adalah Siklus Hidup Pengembangan Sistem atau lebih dikenal dengan istilah SDLC. SDLC meliputi serangkaian langkah yang dimulai dari Tahap Perencanaan, Analisis, Perancangan, implementasi, hingga pemeliharaan sistem. Konsep SDLC ini menjadi dasar bagi beragam model pengembangan perangkat lunak yang membentuk kerangka kerja untuk merencanakan dan

mengendalikan proses pembuatan sistem informasi. SDLC memiliki beberapa model dalam implementasi tahapan prosesnya, termasuk Model Sequential atau Waterfall, Model Paralel, Model Iteratif, Model Prototipe, Model RAD (Rapid Application Development), Model Spiral, Model V-Shaped, dan Pengembangan Agile.

## 2.2 Teori Khusus

### 2.2.1 *Sistem Development Life Cycle (SDLC)*

*Sistem Development Life Cycle (SDLC)* merupakan sebuah metodologi atau pendekatan yang digunakan oleh organisasi untuk mengembangkan dan mendukung sistem informasi mereka. SDLC membantu dalam merencanakan, merancang, membangun, menguji, dan mengelola sistem informasi dengan cara yang terstruktur dan terorganisir (Valacich & George, 2017). Pada dasarnya, SDLC adalah suatu proses untuk membangun sistem baru atau mengembangkan sistem yang sudah ada yang di dalamnya terdapat langkah-langkah yang harus diikuti dalam mengembangkan sistem informasi dari awal hingga akhir. SDLC terdiri dari empat tahap dasar yaitu perencanaan, analisis, desain, dan implementasi. Meskipun setiap proyek dapat menekankan atau mendekati fase SDLC dengan cara yang berbeda, semua proyek memiliki elemen dari keempat tahap ini. Masing-masing tahap memiliki rangkaian langkah-langkah yang bergantung pada teknik tertentu untuk menghasilkan keluaran berupa dokumen dan file yang memberikan pemahaman yang lebih dalam tentang proyek tersebut (Dennis, Wixom, & Tegarden, 2015). Pada gambar 2.2. merupakan fase tahapan dalam SDLC.



Gambar 2.2 *Sistem Development Life Cycle*  
Sumber : (Dennis, Wixom, & Tegarden)



Pada gambar 2.2. terlihat bahwa dari siklus hidup tersebut setiap fase terlihat seperti dilakukan secara berurutan, namun sebenarnya tidak, setiap fase yang dilakukan dimaksudkan untuk disesuaikan kembali dengan kebutuhan proyek. Dalam setiap fase SDLC, proyek dapat kembali ke fase sebelumnya jika diperlukan. Dalam SDLC, dimungkinkan juga untuk menyelesaikan beberapa aktivitas dalam satu fase secara bersamaan dengan beberapa aktivitas di fase lainnya (Valacich & George, 2017).

Untuk lebih jelasnya berikut merupakan penjelasan dari setiap tahapan SDLC :

- *Planning*

Fase ini merupakan tahap proses yang mendasar untuk memahami mengapa suatu sistem informasi harus dibangun dan menetapkan strategi pembangunan oleh tim proyek. Tahap ini terdiri dari dua langkah utama:

1. Inisiasi Proyek dan Penilaian Nilai Bisnis

Pada tahap inisiasi proyek, nilai bisnis sistem bagi organisasi diidentifikasi, seperti bagaimana sistem dapat mengurangi biaya atau meningkatkan pendapatan. Banyak ide untuk sistem baru berasal dari luar divisi Sistem Informasi (contohnya dari departemen pemasaran atau akuntansi) dalam bentuk permintaan sistem. Permintaan sistem tersebut memberikan ringkasan singkat tentang kebutuhan bisnis dan menjelaskan bagaimana sistem yang mendukung kebutuhan tersebut akan memberikan nilai bisnis. Departemen Sistem Informasi bekerja sama dengan pihak atau departemen yang mengajukan permintaan (disebut sebagai sponsor proyek) untuk melakukan analisis kelayakan. Permintaan sistem dan analisis kelayakan ini kemudian disajikan kepada komite persetujuan Sistem Informasi (sering kali disebut sebagai komite pengarah), yang bertugas memutuskan apakah proyek tersebut harus dilanjutkan.

2. Manajemen Proyek

Setelah proyek disetujui, proyek tersebut memasuki tahap manajemen proyek. Selama manajemen proyek, manajer proyek merencanakan kerja, mengatur proyek, dan menerapkan teknik yang membantu tim proyek mengendalikan serta mengarahkan proyek melalui seluruh SDLC. Hasil dari manajemen proyek adalah rencana proyek yang menggambarkan bagaimana tim proyek akan mengembangkan sistem.

- *Analysis*

Fase ini dalam pengembangan sistem merupakan tahap untuk menentukan tentang siapa yang akan menggunakan sistem, apa yang akan dilakukan oleh sistem, dan di mana serta kapan sistem tersebut akan digunakan. Pada tahap ini, tim proyek melakukan penyelidikan terhadap sistem yang sudah ada, mengidentifikasi peluang perbaikan, dan mengembangkan konsep untuk sistem yang baru. Fase ini terdiri dari tiga langkah utama:

1. Strategi Analisis

Pembuatan strategi ini bertujuan untuk memandu usaha tim proyek. Biasanya, strategi ini melibatkan analisis sistem yang sedang berjalan (disebut sistem apa adanya) dan masalah yang ada, serta cara merancang sistem yang baru (disebut sistem calon).

2. Pengumpulan Persyaratan

Langkah ini melibatkan pengumpulan persyaratan melalui wawancara atau kuesioner. Informasi dari proses ini, bersama dengan masukan dari sponsor proyek dan berbagai pihak, membentuk dasar untuk mengembangkan konsep sistem baru. Konsep ini menjadi landasan untuk mengembangkan serangkaian model analisis bisnis, yang menggambarkan operasi bisnis jika sistem baru dikembangkan.

3. Penggabungan Hasil Analisis

Hasil dari analisis, konsep sistem, dan model disatukan dalam sebuah dokumen yang disebut usulan sistem. Dokumen ini disajikan kepada sponsor proyek dan pemangku kepentingan penting lainnya, yang akan menentukan apakah proyek tersebut akan dilanjutkan. Usulan sistem ini memberikan gambaran awal tentang persyaratan bisnis yang harus dipenuhi oleh sistem baru.

- *Design*

Fase ini dalam pengembangan sistem merupakan tahap dalam menentukan bagaimana sistem akan beroperasi, baik dari segi perangkat keras, perangkat lunak, dan infrastruktur jaringan; antarmuka pengguna, formulir, dan laporan; hingga program, database, dan file yang spesifik diperlukan. Fase ini memiliki empat langkah utama:

1. Strategi Desain

Langkah pertama dalam fase desain ini adalah mengembangkan strategi desain yang menjelaskan bagaimana sistem akan dikembangkan, apakah oleh tim internal, dialihdayakan ke perusahaan lain, atau membeli paket perangkat lunak yang ada.

2. Perancangan Arsitektur

Tahap ini melibatkan pengembangan dasar desain arsitektur untuk sistem yang menggambarkan perangkat keras, perangkat lunak, dan infrastruktur jaringan yang akan digunakan.

3. Spesifikasi Database dan File

Langkah ini menentukan dengan tepat data apa yang akan disimpan dan di mana data tersebut akan disimpan.

4. Pengembangan Desain Program

Tim analis mengembangkan desain program yang mendefinisikan program-program yang perlu ditulis dan apa yang akan dilakukan oleh setiap program.

Kumpulan hasil dari fase desain ini (desain arsitektur, antarmuka, spesifikasi *database* dan file, serta desain program) diserahkan kepada tim pemrograman untuk diimplementasikan. Pada akhir fase ini, analisis kelayakan dan rencana proyek direvisi, dan keputusan terkait kelanjutan proyek dibuat oleh sponsor proyek dan komite persetujuan.

- *Implementation*

Fase ini adalah tahap terakhir dalam SDLC di mana sistem aktual dibangun atau dibeli (untuk perangkat lunak yang dikemas). Ini adalah fase yang sering kali memerlukan perhatian khusus karena proses pengembangan sistem yang paling lama dan mahal. Tahap ini memiliki tiga langkah utama:

1. Konstruksi Sistem

Langkah pertama adalah membangun dan menguji sistem untuk memastikan kinerjanya sesuai dengan desainnya. Pengujian dilakukan dengan cermat karena bug dapat memiliki dampak besar pada kinerja sistem.

2. Instalasi Sistem

Instalasi merupakan proses di mana sistem lama dimatikan dan sistem baru diaktifkan. Pengembangan rencana pelatihan penting untuk mengajarkan pengguna cara menggunakan sistem baru dan membantu mengelola perubahan yang dibawa oleh sistem baru.

3. Rencana Dukungan Sistem

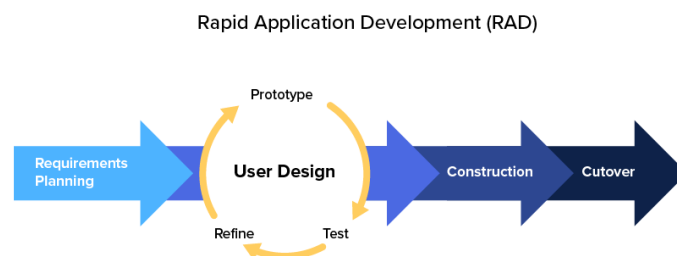
Tim analis membuat rencana dukungan untuk sistem, yang meliputi tinjauan pasca-implementasi serta strategi untuk mengidentifikasi perubahan yang diperlukan untuk sistem.

### **2.2.2 Metode Pengembangan Rapid Application Development (RAD)**

Rapid Application Development (RAD) merupakan sebuah pendekatan pengembangan yang menekankan pada pembuatan prototipe dengan cepat dan siklus umpan balik yang sering selama tahapan

pengembangan dan pengujian yang memakan waktu. Dengan mengadopsi pendekatan ini, pengembang dapat melaksanakan beberapa iterasi dan pembaruan terhadap perangkat lunak dengan cepat, tanpa harus memulai proses pengembangan dari awal setiap kali. Model RAD muncul sebagai alternatif saat para pengembang menyadari bahwa model pengembangan waterfall tradisional tidak selalu efisien (Saputra & Aprilian, 2020 p. 63).

Husain (2023) menjelaskan terdapat 4 fase dalam pengembangan dengan metode RAD di antaranya adalah *Requirements Planning*, *User Design*, *Rapid Construction*, dan *custover*. Pada gambar 2 menampilkan tahapan pada metode RAD.



*Gambar 2.3 Rapid Application Development (RAD) methodology*  
*sumber : (Husain 2023)*

#### 1. Requirements Planning

Tahap ini berhubungan dengan pertemuan lokasi proyek. Dibandingkan dengan metode manajemen proyek lainnya, tahap desain singkat, namun ini merupakan langkah penting untuk keberhasilan proyek sepenuhnya. Pada fase ini, pengembang, pelanggan (pengguna perangkat lunak) dan anggota tim berkomunikasi bersatu satu sama lain untuk menetapkan tujuan dan harapan proyek, serta untuk memecahkan masalah saat ini dan potensi masalah selama proses pembuatan

#### 2. User Design

Setelah desain ditentukan, dilanjutkan dengan pengembangan dan membangun desain sistem melalui iterasi prototipe yang berbeda. Inilah inti dari pendekatan RAD yang membedakannya dengan strategi manajemen proyek lainnya. Pada tahap ini, klien bekerja dengan

pengembang untuk memastikan bahwa persyaratan mereka terpenuhi pada setiap langkah proses desain. Sama seperti pengembangan perangkat lunak yang dapat disesuaikan, ini memungkinkan pengguna menguji prototipe produk apa pun pada langkah apa pun untuk memastikan bahwa produk tersebut memenuhi harapan mereka. Semua kesalahan dan node diperbaiki dalam proses berulang. Pengembang merancang prototipe, pelanggan (pengguna) mengujinya, dan kemudian mereka bertemu untuk mendiskusikan efektivitas dan tidak efektifan metode tersebut. Metode ini memungkinkan pengembang untuk memodifikasi model hingga mereka puas dengan desainnya. Pengembang perangkat lunak dan pelanggan dapat belajar dari pengalaman dan memastikan bahwa hal-hal tertentu tidak akan tergelincir.

### 3. Rapid Construction

Fase ketiga ini tahap yang penting karena klien tetap memberikan masukan sepanjang proses. Mereka dapat menyarankan perubahan, perubahan atau bahkan ide-ide baru yang dapat memecahkan masalah yang muncul. Pada tahap ketiga diperoleh prototype dan sistem Beta dari tahap desain dan ditransformasikan menjadi model kerja. Sebagian besar masalah dan perubahan diselesaikan dalam fase desain berulang yang lengkap, sehingga memungkinkan pengembang untuk membuat model kerja akhir lebih cepat dibandingkan metode manajemen proyek tradisional

### 4. Custover

Custover (Peralihan) adalah fase perakitan ketika produk akhir dirilis. Hal ini mencakup transfer data, pengujian, transisi ke sistem baru, dan pelatihan pengguna. Semua perubahan akhir dilakukan saat pengembang dan pelanggan terus memeriksa kesalahan sistem.

## 2.2.3 *Unified Modelling Language (UML)*

Dennis (2015) menjelaskan *Unified Modeling Language (UML)* merupakan sebuah standar industri yang dikembangkan pada tahun 1995

oleh Grady Booch, Ivar Jacobson, James Rumbaugh, dan tim lainnya. UML bertujuan untuk menyediakan bahasa umum dan teknik diagram yang kaya untuk memodelkan sistem berorientasi objek dari analisis hingga implementasi. Pada tahun 1997, UML diadopsi sebagai standar oleh Object Management Group (OMG).

Untuk UML versi 2.5 terdapat lima belas teknik diagram yang digunakan untuk memodelkan sistem, dibagi menjadi dua kelompok besar. *Structure Diagrams* untuk merepresentasikan data dan hubungan statis dalam sistem, dan *Behavioral Diagrams* untuk menggambarkan hubungan dinamis antara objek dan perilaku dinamis objek tersebut. *Structure Diagrams* mencakup berbagai jenis diagram seperti kelas, objek, dan lainnya, sementara *Behavioral Diagrams* termasuk aktivitas, urutan, komunikasi, dan lainnya. Pada tabel 2.1 merupakan Ringkasan mengenai *Structure Diagrams* dan *Behavioral Diagrams*.

Tabel 2.1 Ringkasan Diagram UML

Nama Diagram	Fungsi	Fase Primer
<b><i>Structure Diagrams</i></b>		
Class	Mengilustrasikan hubungan antar kelas yang dimodelkan di dalam sistem	Analysis, Design
Object	Mengilustrasikan hubungan antar objek yang dimodelkan dalam sistem; digunakan ketika <i>instance</i> kelas yang sebenarnya akan mengkomunikasikan model dengan lebih baik	Analysis, Design
Package	Mengelompokkan elemen UML lainnya untuk membentuk konstruksi ke tingkat yang lebih spesifik	Analysis, Design, Implementation
Deployment	Digunakan untuk menunjukkan	Physical Design,

	komponen perangkat lunak yang sedang dikerahkan ke arsitektur fisik.	Implementation
Component	Mengilustrasikan hubungan fisik antar komponen perangkat lunak	Physical Design, Implementation
Composite Structure Design	Menggambarkan struktur internal suatu kelas	Analysis, Design
Profile	Digunakan untuk mengembangkan ekstensi UML itu sendiri	-
<b><i>Behavioral Diagrams</i></b>		
Activity	Mengilustrasikan alur kerja bisnis yang tidak bergantung pada <i>class</i> , alur aktivitas dalam <i>use case</i> , atau desain metode secara mendetail	Analysis, Design
Sequence	Menggambarkan model perilaku objek dalam <i>use case</i> berfokus pada pengurutan aktivitas berdasarkan waktu	Analysis, Design
Communication	Memodelkan perilaku objek dalam <i>use case</i> berfokus pada komunikasi di antara sekumpulan objek yang berkolaborasi dalam suatu kegiatan	Analysis, Design
Interaction Overview	Memberikan gambaran umum tentang aliran kendali suatu objek	Analysis, Design
Timing	Mengilustrasikan interaksi antara sekumpulan objek dan keadaan perubahan yang mereka lalui sepanjang sumbu waktu	Analysis, Design
Behavioral State Machine	Memeriksa perilaku suatu <i>class</i>	Analysis, Design
Protocol State Machine	Mengilustrasikan ketergantungan di antara berbagai antarmuka suatu	Analysis, Design



	kelas	
Use-Case	Menangkap kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dengan lingkungannya	Analysis,

*Sumber : System Analysis & Design An Object-Oriented Approach with UML, 2015*

Berdasarkan permodelan diagram yang ada di UML berikut merupakan penjelasan mengenai diagram dan notasi-notasi yang akan digunakan untuk mengembangkan sistem :


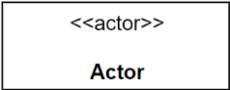
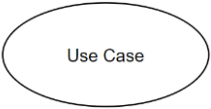
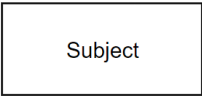

a. *Use case* Diagram

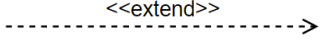
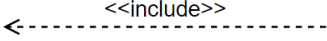

*Use case* diagram merupakan cara yang sederhana dan lugas untuk menggambarkan fungsionalitas utama sistem serta berbagai jenis pengguna yang akan berinteraksi dengan sistem tersebut. *Use case* diagram membantu dalam komunikasi dengan pengguna dan pemangku kepentingan untuk memahami apa yang sebenarnya akan dilakukan oleh sistem

(Dennis, Wixom, & Tegarden, 2015). Diagram ini digunakan saat mengumpulkan dan mendefinisikan persyaratan untuk sistem, dan dapat membuat pengguna untuk memberikan persyaratan tingkat tinggi tambahan. *Use case* diagram memberikan gambaran tingkat tinggi tentang interaksi antara aktor (pengguna) dan fungsi-fungsi sistem yang akan digunakan oleh mereka. Pada tabel 2.2 menjelaskan aturan penggunaan notasi dan sintaksis untuk *use case* diagram.

Elemen *use case* diagram meliputi *actor*, *use case*, *subject boudaries*, dan sekumpulan hubungan antara *actor* dengan *use case*. Hubungan tersebut terdiri atas hubungan *association*, *include*, *extend*, dan *generalization*. Pada tabel 2.2 menjelaskan aturan penggunaan notasi dan sintaksis untuk *use case* diagram.

Tabel 2.2 Sintaksis dan notasi Use case Diagram

Notasi	Nama Notasi	Keterangan
 Actor 	Actor	<ul style="list-style-type: none"> <li>Pengguna atau sistem yang memperoleh manfaat atau berinteraksi dengan sistem</li> <li>Digambarkan dengan simbol orang atau jika ada aktor yang bukan maka digambarkan dengan simbol persegi panjang dengan keterangan &lt;&lt;actor&gt;&gt; di dalamnya (alternatif).</li> <li>Diberi label keterangan berdasarkan perannya</li> <li>Dapat dikaitkan dengan <i>actor lain</i> menggunakan <i>Inheritance/specialization</i> digambarkan dengan panah dengan mata panah berongga.</li> <li>Ditempatkan di luar <i>subject boundary</i></li> </ul>
	Use case	<ul style="list-style-type: none"> <li>Mewakili bagian utama dari fungsionalitas sistem</li> <li>Dapat digunakan untuk memperluas (Extend) <i>use case</i> lainnya.</li> <li>Dapat menyertakan (Include) <i>use case</i> lainnya</li> <li>Ditempatkan di dalam batas sistem</li> <li>Dilabelkan dengan frasa kata kerja / kata benda deskriptif</li> </ul>
	Boundary	<ul style="list-style-type: none"> <li>Dalam menggambarkan <i>subject boundary</i> nama subjek diletakan di dalam atau di atas notasi</li> <li><i>Subject boundary</i> mewakili ruang lingkup subjek, misalnya sistem atau proses individu</li> </ul>
	Association Relationship	<ul style="list-style-type: none"> <li>Menghubungkan <i>actor</i> dengan <i>use case</i> yang berinteraksi dengannya</li> </ul>

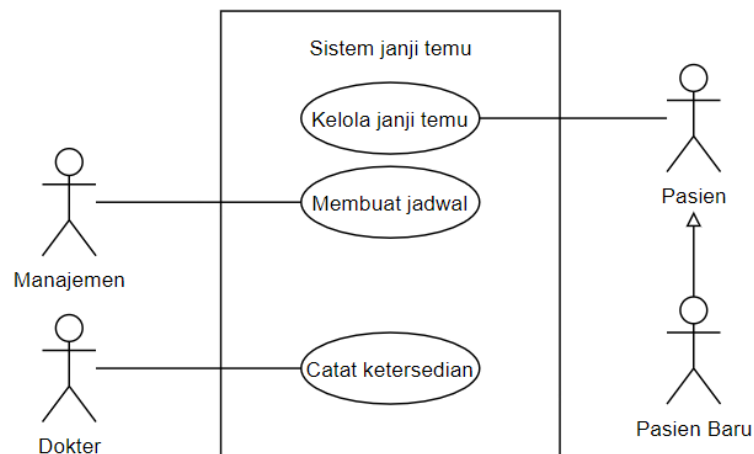
	<p><i>Extend Relationship</i></p>	<ul style="list-style-type: none"> <li>• Mewakili perluasan <i>use case</i> untuk menyertakan fungsi atau perilaku opsional</li> <li>• Digambarkan dengan simbol panah yang ditarik dari <i>use case</i> ekstensi ke <i>use case</i> dasar</li> </ul>
	<p><i>Include Relationship</i></p>	<ul style="list-style-type: none"> <li>• Merupakan penyertaan fungsionalitas dari <i>use case</i> ke <i>use case</i> lainnya</li> <li>• Digambarkan dengan simbol panah dengan garis putus-putus yang ditarik dari <i>use case</i> dasar ke <i>use case</i> yang digunakan</li> </ul>
	<p><i>Generalization Relationship</i></p>	<ul style="list-style-type: none"> <li>• Mewakili <i>use case</i> khusus ke <i>use case</i> yang lebih umum</li> <li>• Dapat digunakan sebagai pengait pada <i>actor</i> yang memiliki sub <i>actor</i> atau turunan <i>actor</i> (inheritence)</li> <li>• Digambarkan dengan simbol panah dengan mata panah berongga yang ditarik dari <i>use case</i> khusus ke <i>use case</i> dasar</li> </ul>

Sumber : (Dennis, Wixom, & Tegarden, 2015)

Untuk lebih jelasnya (Dennis, Wixom, & Tegarden, 2015) menjelaskan detail dari setiap notasi dan cara menggambarannya ke dalam sebuah *use case* diagram :

### **Actor**

*Actor* dalam diagram *use case* merupakan representasi dari peran atau perangkat dalam lingkungan yang berinteraksi dengan sistem, bukan pengguna individu. *Actor* dapat mewakili pengguna sistem atau bahkan sistem lain yang berinteraksi dengan sistem saat ini. *Actor* bisa memberikan input kepada sistem, menerima output dari sistem, atau melakukan keduanya. Sebagai contoh pada gambar 2.4 tiga aktor berinteraksi pada sebuah sistem janji temu (pasien, dokter, dan manajemen).



Gambar 2.4. Diagram Use case Sistem janji temu (Dennis, Wixom, & Tegarden, 2015)

### Subject Boundary

*Subject boundary* atau batasan subjek dalam use case diagram merupakan sebuah notasi yang dimodelkan dengan bentuk kotak, digunakan untuk menggambarkan ruang lingkup sistem dengan jelas, baik yang bersifat eksternal maupun internal, seperti yang diperlihatkan pada tabel 2.2. Salah satu keputusan penting adalah menentukan batasan subjek. Hal tersebut dapat digunakan untuk memisahkan perangkat lunak dari lingkungan eksternalnya, memisahkan sub sistem dalam perangkat lunak, atau bahkan memisahkan proses individual dalam sistem. Batasan subjek juga bisa digunakan untuk memisahkan sistem informasi, termasuk perangkat lunak dan aktor internal, dari lingkungannya.

### Inheritance

Terkadang, ada situasi di mana seorang *actor* memainkan peran yang sedikit berbeda dari *actor* umumnya. Seperti yang terlihat pada gambar 2.4. Dalam kasus tersebut yang ada pada gambar 2.4 *actor* Pasien Baru merupakan *actor* turunan dari *actor* Pasien, hal ini merupakan contoh dari penggunaan konsep *inheritence* atau *generalization*. Dengan penggunaan *inheritance* pada aktor dalam *use case* diagram tersebut,

menandakan bahwa Pasien baru adalah turunan atau sub aktor dari Pasien. Dengan demikian, Pasien Baru akan mewarisi semua *use case* yang dimiliki oleh Pasien, termasuk "Kelola Janji Temu." Hal ini berarti bahwa baik Pasien maupun Pasien Baru akan memiliki akses ke *use case* "Kelola Janji Temu" dan mampu menggunakannya, biasanya dimodelkan dengan garis dengan segitiga kosong di ujung model.

### **Association**

Hubungan antara *use case* dan *actor* dalam *use case* diagram diwakili oleh *association*, yang menunjukkan bagaimana *use case* berinteraksi dengan aktor, seperti yang diperlihatkan di Gambar 2.4. Garis yang menghubungkan *actor* dengan *use case* mewakili *association*, yang umumnya mencerminkan komunikasi dua arah antara Kasus Penggunaan dan *actor*. Jika komunikasi hanya satu arah, maka panah padat dapat digunakan untuk menunjukkan arah aliran informasi.

Sebagai contoh dalam Gambar 2.4, *actor* Pasien berhubungan dengan *use case* "Melola Janji Temu." Karena tidak ada panah pada hubungan ini, komunikasinya bersifat dua arah.

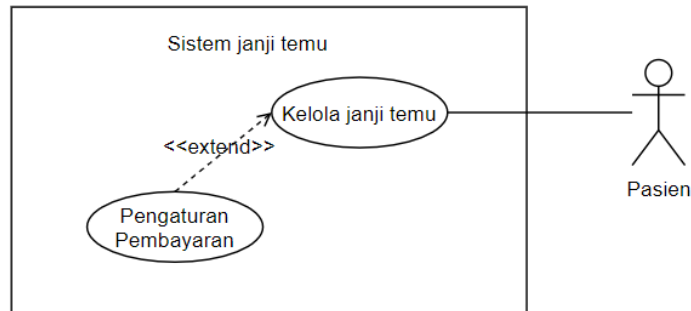
### **Use case**

*Use case* (kasus penggunaan) dalam UML digambarkan dengan bentuk oval, dapat dilihat pada tabel 2.2. *Use case* mewakili proses utama yang dilakukan oleh sistem, memberikan manfaat bagi aktor atau pemangku kepentingan tertentu. Setiap *use case* diberi label dengan frasa kata kerja-kata benda deskriptif yang menjelaskan tindakan yang dilakukan dalam *use case* tersebut. Dalam beberapa situasi, sebuah *use case* dapat menggabungkan, memperluas, atau menggeneralisasi fungsionalitas dari *use case* lain dalam diagram.

### **Extend Relationship**

Hubungan "Extend" adalah salah satu jenis hubungan yang digunakan dalam Use Case Diagram (diagram kasus penggunaan) untuk menggambarkan bagaimana sebuah *use case* bisa memperpanjang atau mengekstensi perilaku dari *use case* lain (Dennis, Wixom, & Tegarden,

2015). Hubungan "Extend" digunakan untuk menggambarkan skenario opsional atau alternatif yang bisa terjadi dalam suatu *use case*. Hal ini membantu dalam menjelaskan bagaimana suatu *use case* dapat melengkapi atau memperluas *use case* lainnya dalam reaksi terhadap kondisi tertentu. Sebagai contoh pada gambar 2.5 merupakan contoh penggunaan *Extend Relationship*.



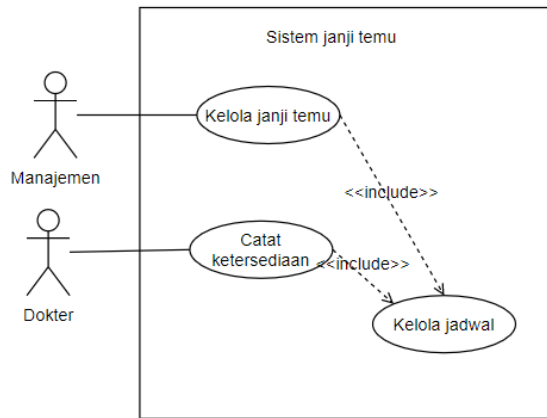
Gambar 2.5. Contoh penggunaan *Extend Relationship* (Dennis, Wixom, & Tegarden, 2015)

Jika dalam *use case* "Kelola Janji Temu" terdapat kebutuhan untuk memverifikasi atau membuat pengaturan pembayaran, maka kita dapat membuat *use case* terpisah yang disebut "Pengaturan Pembayaran" yang memperluas fungsionalitas "Kelola Janji Temu." Hubungan ini ditunjukkan dengan panah yang berlabel "<<extend>>" dari "Pengaturan Pembayaran" ke "Kelola Janji Temu" dalam diagram *use case*. Posisi vertikal *use case* dalam diagram juga dapat memberikan hierarki dan tingkat detail yang lebih tinggi atau lebih rendah dalam fungsionalitas sistem.

### **Include Relationship**

Dalam beberapa situasi, satu *use case* dapat mengandung fungsi-fungsi umum yang digunakan oleh *use case* lainnya. Contohnya, ada kasus penggunaan bernama "Kelola Jadwal" yang melibatkan beberapa tugas rutin yang dibutuhkan untuk mengelola jadwal janji temu dokter. Dua *use case* lain, yaitu "Catat Ketersediaan" dan "Membuat Jadwal," keduanya juga melibatkan tugas-tugas ini. Gambar 2.6 menunjukkan cara merancang

sistem agar "Kelola Jadwal" menjadi *use case* bersama yang digunakan oleh *use case* lainnya.

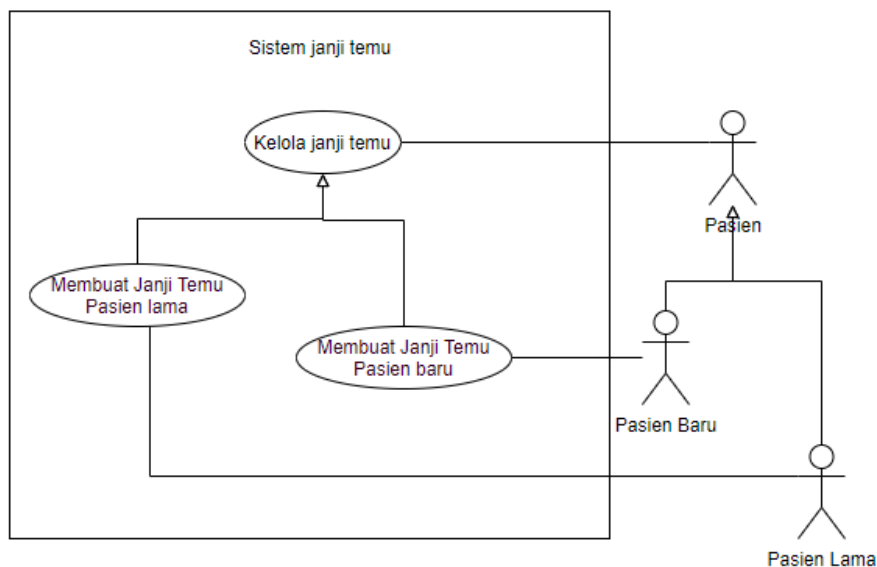


Gambar 2.6. Contoh penggunaan *Include Relationship* (Dennis, Wixom, & Tegarden, 2015)

Hubungan *include* ditunjukkan dengan panah berlabel "<<include>>" dan *use case* yang termasuk digambarkan di bawah *use case* yang mengandungnya. Dalam contoh ini, "Catat Ketersediaan " dan "Membuat Jadwal " menggunakan fungsionalitas yang disertakan dari "Kelola Jadwal".

### **Generalization Relationship**

Terakhir, ada situasi di mana kita dapat menggunakan hubungan generalisasi untuk menyederhanakan *use case* individual. Sebagai contoh, dalam Gambar 2.7, *use case* "Kelola Janji Temu" telah dikhususkan untuk mencakup *use case* untuk Pasien Lama dan Pasien Baru.



Gambar 2.7. Contoh penggunaan Generalization Relationship (Dennis, Wixom, & Tegarden, 2015)

Pada gambar 2.7 *use case* "Membuat janji temu pasien lama" mewarisi fungsionalitas dari "Kelola Janji Temu". *Use case* "Membuat janji temu Pasien Baru" juga mewarisi semua fungsi dari "Kelola Janji Temu" yang umum dan memanggil *use case* "Membuat janji temu pasien baru," yang mencakup fungsi untuk memasukkan Pasien Baru ke dalam database. Hubungan generalisasi diwakili sebagai panah berongga tanpa label, dengan *use case* yang lebih umum berada di tingkat hierarki yang lebih tinggi daripada *use case* yang lebih khusus. Aktor juga dapat diperluas dengan cara yang serupa, seperti dalam contoh dengan Pasien Lama dan Pasien Baru.

b. *Use Case Description*

*Use case description* merupakan metode formal untuk mendokumentasikan sebuah kasus penggunaan. Dokumentasi ini dibuat untuk meningkatkan pemahaman pengguna tentang keseluruhan proses bisnis dan bagaimana sistem mendukungnya, dengan tujuan menciptakan sistem yang komprehensif yang sesuai dengan kebutuhan (Satzinger, Jackson, & Burd, 2016). Metode ini membantu dalam menjelaskan bagaimana sebuah sistem berinteraksi dengan pengguna untuk menyelesaikan tugas atau fungsionalitas tertentu, memastikan pemahaman



yang mendalam tentang fungsionalitas sistem dan bagaimana hal tersebut sesuai dengan kebutuhan pengguna dalam suatu skenario atau proses tertentu. Pada tabel 2.3 merupakan contoh *use case description* yang dikembangkan dari *use case* “Buat Akun Pelanggan”.

Tabel 2.3 contoh *use case description*

<b>Use case name:</b>	Buat akun pelanggan.	
<b>Scenario:</b>	Buat akun pelanggan online.	
<b>Triggering event:</b>	Pelanggan baru ingin membuat akun secara online	
<b>Brief description:</b>	Pelanggan online membuat akun pelanggan dengan memasukkan informasi dasar dan kemudian menindaklanjuti dengan satu atau lebih alamat dan kartu kredit atau debit.	
<b>Actors:</b>	Pelanggan	
<b>Related use cases:</b>	-	
<b>Stakeholders:</b>	Accounting, Marketing, Sales.	
<b>Preconditions:</b>	Subsistem Akun Pelanggan harus tersedia. Layanan otorisasi kredit/debit harus tersedia.	
<b>Postconditions:</b>	Pelanggan harus dibuat dan disimpan. Satu atau lebih Alamat harus dibuat dan disimpan. Informasi kartu kredit/debit harus divalidasi. Akun harus dibuat dan disimpan. Alamat dan Rekening harus dikaitkan dengan Pelanggan.	
<b>Flow of activities:</b>	<b>Actor</b>	<b>System</b>
	1. Pelanggan ingin membuat akun pelanggan dan memasukkan informasi dasar pelanggan.	1.1 Sistem menciptakan pelanggan baru.
		1.2 Perintah sistem untuk pelanggan memasukkan alamat.
	2. Pelanggan memasukkan satu atau lebih alamat.	2.1 Sistem membuat alamat.
	3. Pelanggan memasukkan kartu kredit/debit informasi.	2.2 Sistem meminta kredit/debit kartu.
		3.1 Sistem membuat akun.
		3.2 Sistem memverifikasi otorisasi untuk kartu

		kredit/debit. 3.3 Sistem mengasosiasikan pelanggan, alamat, dan akun. 3.4 Sistem mengembalikan pelanggan yang valid Detail akun.
<b>Exception conditions:</b>	1.1 Data dasar pelanggan tidak lengkap. 2.1 Alamatnya tidak valid. 3.2 Informasi kredit/debit tidak valid.	

Sumber : (Satzinger, Jackson, & Burd, 2016)

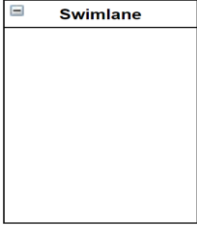


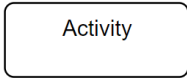

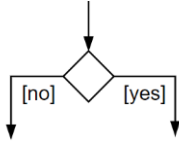
Tabel 2.3 berfungsi sebagai template standar untuk mendokumentasikan deskripsi yang dikembangkan untuk kasus pengguna.

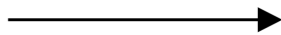
#### c. Activity Diagram

Dalam mendokumentasikan rincian *use case*, salah satu metode yang sering digunakan adalah menggunakan *activity* diagram atau diagram aktivitas. *Activity* diagram digunakan untuk memodelkan berbagai proses yang terlibat dalam suatu *use case*. Tujuan dari permodelan ini adalah untuk mendokumentasikan cara interaksi antara pengguna dan sistem, serta bagaimana pengguna dapat berinteraksi dengan sistem untuk menyelesaikan aktivitas atau tugas tertentu dalam satu *use case* (Satzinger, Jackson, & Burd, 2016).

*Activity* diagram membantu dalam menggambarkan urutan langkah atau prosedur yang dilakukan oleh pengguna atau sistem untuk mencapai tujuan yang diinginkan dalam skenario *use case* tersebut. Berikutnya pada tabel 2.4 merupakan notasi dasar yang digunakan dalam menggambarkan *activity* diagram.

Tabel 2.4 Notasi Activity Diagram

Notasi	Nama Notasi	Keterangan
	<i>Swimlane</i>	<ul style="list-style-type: none"> <li>Mewakili aktor yang melakukan aktivitas-aktivitas</li> <li>Jika terdapat beberapa <i>swimlane</i> digambarkan sejajar dengan <i>swimlane</i> lainnya.</li> </ul>
	<i>Initial Node</i>	<ul style="list-style-type: none"> <li>Mewakili alur awal berjalannya sistem dimulai.</li> <li>Hanya memiliki satu garis untuk berinteraksi dengan <i>activity</i></li> </ul>
	<i>Final Node</i>	<ul style="list-style-type: none"> <li>Mewakili berakhirnya alur sistem atau aktivitas.</li> </ul>
	<i>Activity</i>	<ul style="list-style-type: none"> <li>Mewakili aktivitas-aktivitas individu dalam alur kerja</li> </ul>
	<i>Synchronization bar</i>	<ul style="list-style-type: none"> <li>Membagi jalur menjadi beberapa jalur konkuren (Split) atau menggabungkan jalur konkuren (Join)</li> <li>Digambarkan dengan simbol garis tebal dan solid</li> </ul>
	<i>Decision activity</i>	<ul style="list-style-type: none"> <li>mewakili aktivitas-aktivitas individu dalam alur kerja.</li> <li>Digambarkan dengan bentuk belah ketupat / berlian</li> </ul>

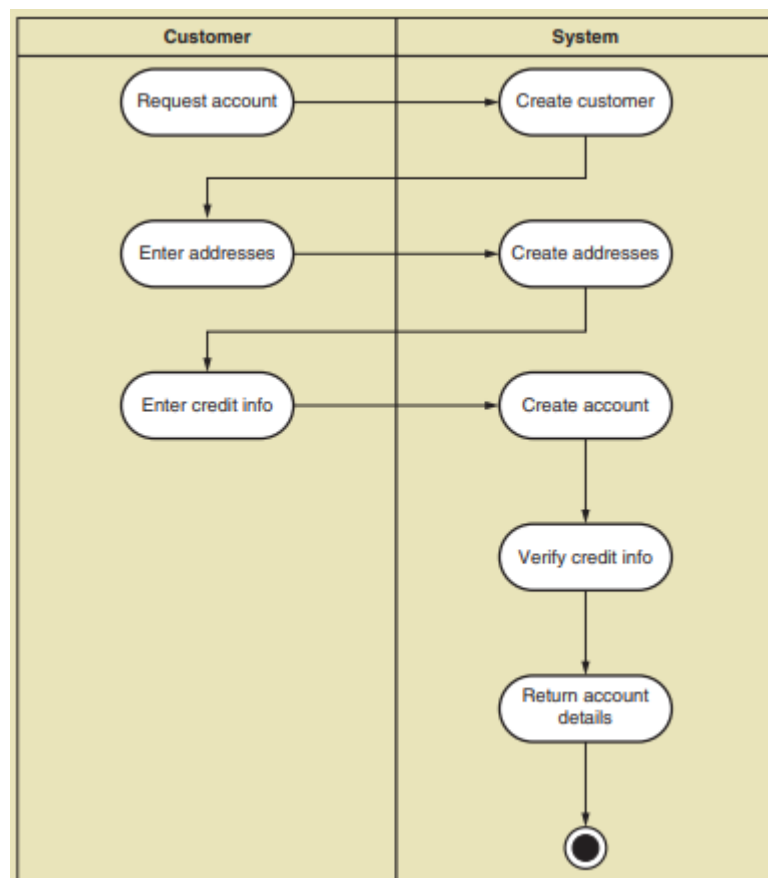


Transition arrow

- Sebagai penghubung yang mewakili urutan dari setiap *activity*.

Sumber : (Satzinger, Jackson, & Burd, 2016)

Dalam permodelan *activity diagram* terdapat beberapa kasus, terkadang *activity diagram* dapat menggantikan bagian alur aktivitas dalam *use case description* dan terkadang dibuat untuk melengkapi *use case description* (Satzinger, Jackson, & Burd, 2016, hal. 137). Gambar 2.8 merupakan contoh *activity diagram* untuk *use case* “Buat Akun Pelanggan”. Dalam contoh ini, terdapat dua *swimlane*, satu untuk pelanggan dan satu lagi untuk sistem. Pelanggan memiliki tiga aktivitas, dan sistem memiliki lima aktivitas.

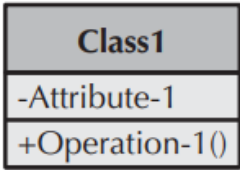




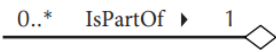
Gambar 2.8 Contoh Activit Diagram (Satzinger, Jackson, & Burd, 2016)

#### d. Class Diagram

*Class Diagram* atau diagram kelas adalah representasi visual yang menggambarkan kelas-kelas dan hubungan yang ada di antara kelas-kelas tersebut yang tetap konsisten dalam sistem sepanjang waktu. Diagram kelas mengilustrasikan kelas-kelas yang mencakup perilaku dan status, serta hubungan yang terbentuk di antara kelas-kelas tersebut. Bagian ini menguraikan unsur-unsur dari diagram kelas, berbagai metode untuk menyederhanakan diagram kelas, dan alternatif lain dari diagram kelas yang disebut diagram objek. Diagram kelas adalah model statis yang membantu dalam memvisualisasikan struktur dasar dari sistem yang terdiri dari objek-objek yang terlibat beserta hubungan antar objek tersebut (Dennis, Wixom, & Tegarden, 2015). Pada tabel 2.5 menunjukkan sintaksis dan notasi dalam *class diagram*.

Tabel 2.5 sintaksis dan notasi class diagram

Notasi/Sintaksis	Nama notasi	Keterangan
	Kelas	<ul style="list-style-type: none"> <li>• Mewakili jenis orang, tempat, atau benda yang diperlukan sistem untuk menangkap dan menyimpan informasi.</li> <li>• Memiliki nama yang diketik tebal dan ditempatkan di tengah pada kompartemen atasnya.</li> <li>• Memiliki daftar atribut di kompartemen tengahnya</li> <li>• Memiliki daftar operasi di kompartemen bawahnya.</li> <li>• Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.</li> </ul>
<i>Attribute name</i>	Atribut	<ul style="list-style-type: none"> <li>• Merupakan properti</li> </ul>

		<ul style="list-style-type: none"> <li>yang menggambarkan keadaan suatu objek.</li> <li>Dapat diturunkan dari atribut lain, ditunjukkan dengan</li> <li>Diberi tanda garis miring di depan nama atribut.</li> </ul>
<i>Operation Name ( )</i>	Operasi	<ul style="list-style-type: none"> <li>Mewakili tindakan atau fungsi yang dapat dilakukan</li> <li>suatu kelas.</li> <li>Dapat diklasifikasikan sebagai konstruktor, kueri, atau operasi pembaruan.</li> <li>Menyertakan tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.</li> </ul>
	Asosiasi	<ul style="list-style-type: none"> <li>Mewakili hubungan antara beberapa kelas atau kelas dan kelas itu sendiri.</li> <li>Bisa ada di antara satu atau lebih kelas.</li> <li>Berisi simbol multiplisitas, yang mewakili angka minimum dan maksimum suatu <i>instance</i> kelas dapat dikaitkan dengan <i>instance</i> kelas terkait.</li> </ul>
	Generalisasi	<ul style="list-style-type: none"> <li>Merupakan semacam hubungan antara beberapa kelas.</li> </ul>
	Agregasi	<ul style="list-style-type: none"> <li>Merupakan bagian logis dari hubungan antara beberapa kelas atau kelas dan kelas itu sendiri.</li> <li>Merupakan suatu</li> </ul>

		bentuk perkumpulan khusus
1..*	IsPartOf ▶	1
	Komposisi	<ul style="list-style-type: none"> <li>• Merupakan bagian fisik dari hubungan antara beberapa kelas atau kelas dan kelas itu sendiri</li> <li>• Merupakan suatu bentuk perkumpulan khusus.</li> </ul>

*Sumber :* (Dennis, Wixom, & Tegarden, 2015)

Untuk lebih jelasnya Dennis (2015) menjelaskan detail dari setiap notasi dan cara menggambarannya ke dalam sebuah *class* diagram :

### **Kelas (*class*)**

Blok penyusun utama *class* diagram adalah kelas, yang menyimpan dan mengelola informasi dalam sistem (lihat tabel 2.5). Selama analisis, kelas mengacu pada orang, tempat, dan hal-hal yang informasinya akan ditangkap oleh sistem. Nantinya, selama desain dan implementasi, kelas dapat merujuk pada artefak implementasi spesifik seperti jendela, formulir, dan objek lain yang digunakan untuk membangun sistem. Setiap kelas digambar menggunakan persegi panjang tiga bagian, dengan nama kelas di atas, atribut di tengah, dan operasi di bawah. Sebagai contoh pada gambar 2.9 dapat dilihat bahwa kelas-kelas yang telah diidentifikasi, Janji Temu, Pasien dan Partisipan. Atribut kelas dan nilainya menentukan *region* setiap objek yang dibuat dari kelas, dan perilakunya diwakili oleh operasi.

Janji Temu	
<ul style="list-style-type: none"> <li>- waktu</li> <li>- Tanggal</li> <li>- catatan</li> </ul>	
+ pembatalan tanpa pemberitahuan()	

Partisipan	pasien
<ul style="list-style-type: none"> <li>- nama keluarga</li> <li>- nama depan</li> <li>- alamat</li> <li>- telepon</li> <li>- tanggal lahir</li> <li>- usia</li> </ul>	<ul style="list-style-type: none"> <li>- jumlah</li> <li>- asuransi</li> </ul>
	<ul style="list-style-type: none"> <li>+ buat janji ()</li> <li>+ hitung kunjungan terakhir ()</li> <li>+ ubah status ()</li> <li>+ memberikan riwayat kesehatan ()</li> </ul>

Gambar 2.9 Contoh penggunaan class, atribut dan operasi  
(Dennis, Wixom, & Tegarden, 2015)

## Atribut

Atribut adalah properti kelas yang informasinya ingin diterima (lihat tabel 2.5). Dapat dilihat pada gambar 2.9 bahwa kelas Partisipan berisi atribut: nama belakang, nama depan, alamat, telepon, dan tanggal lahir dan usia. Pada sebuah atribut dimungkinkan untuk menunjukkan visibilitas atribut pada diagram. Visibilitas berkaitan dengan tingkat penyembunyian informasi yang akan diterapkan pada atribut. Visibilitas suatu atribut dapat bersifat *public* (+), *protected* (#), atau *private* (-).

Atribut *public* adalah atribut yang tidak tersembunyi dari objek lain. Dengan demikian, objek lain dapat mengubah nilainya. Atribut *protected* adalah atribut yang tersembunyi dari semua kelas lain kecuali subkelas terdekatnya. Atribut *private* adalah salah satu yang tersembunyi dari semua kelas lainnya. Visibilitas bawaan untuk suatu atribut biasanya bersifat *private* (Dennis, Wixom, & Tegarden, 2015).

## Operasi

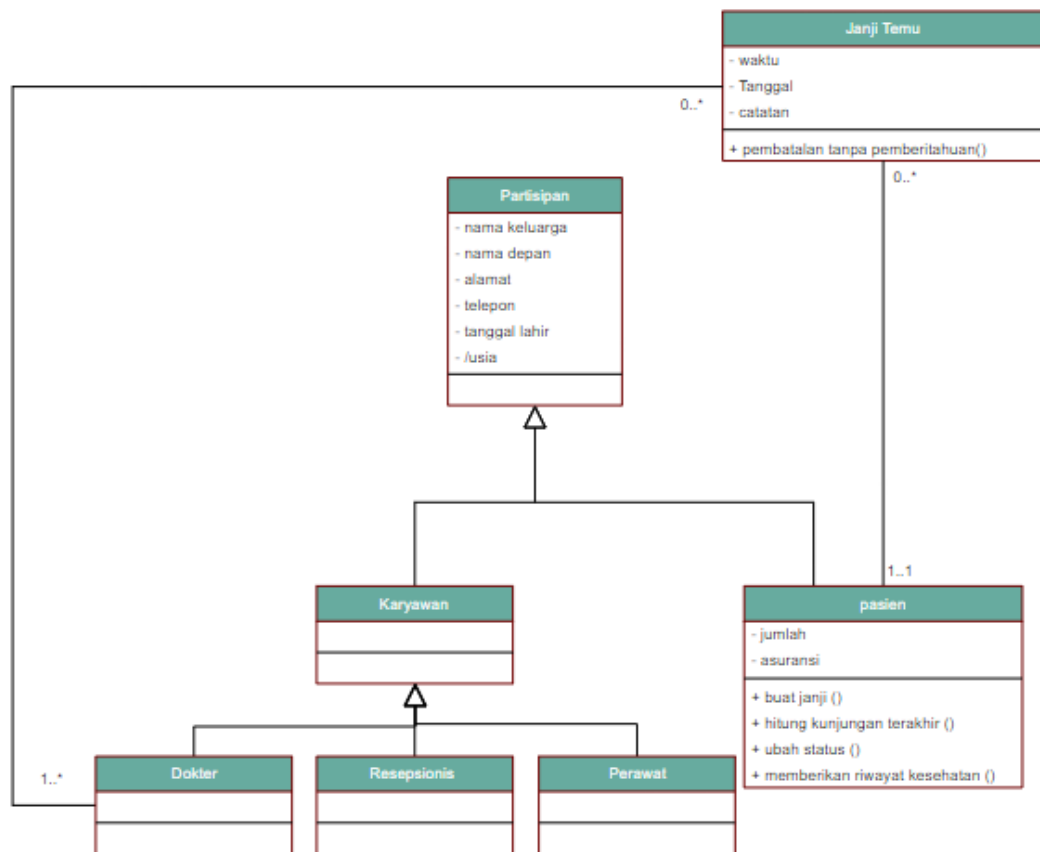
Operasi adalah tindakan atau fungsi yang dapat dilakukan suatu kelas. Fungsi-fungsi yang tersedia untuk semua kelas (misalnya, membuat



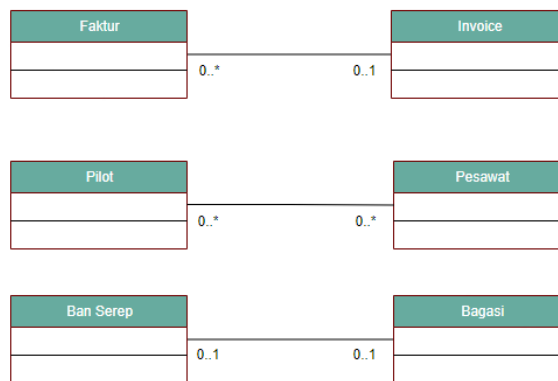
*instance* baru, mengembalikan nilai untuk atribut tertentu, menetapkan nilai untuk atribut tertentu, menghapus sebuah *instance*) tidak secara eksplisit ditampilkan dalam persegi panjang kelas. Sebaliknya, hanya operasi unik untuk kelas tersebut yang disertakan, seperti operasi pembatalan tanpa pemberitahuan di kelas Janji Temu dan operasi penghitungan kunjungan terakhir di kelas Pasien pada Gambar 2.9 dapat dilihat bahwa kedua operasi tersebut diikuti dengan tanda kurung, yang berisi parameter yang diperlukan oleh operasi tersebut. Jika suatu operasi tidak memiliki parameter, tanda kurung tetap ditampilkan tetapi kosong. Seperti halnya atribut, visibilitas suatu operasi dapat ditetapkan sebagai publik, dilindungi, atau pribadi. Visibilitas *default* untuk suatu operasi biasanya bersifat publik.

### **Asosiasi**

Tujuan utama dari *class* diagram adalah untuk menunjukkan hubungan, atau asosiasi, yang dimiliki kelas satu sama lain. Hal ini digambarkan pada diagram dengan menggambar garis antar kelas (lihat tabel 2.5). Ketika beberapa kelas berbagi relasi (atau sebuah kelas berbagi relasi dengan dirinya sendiri), sebuah garis digambar dan diberi label dengan nama relasi atau peran yang dimainkan kelas-kelas tersebut dalam relasi tersebut. Sebagai contoh, pada gambar 2.10 dua kelas pasien dan janji temu dihubungkan satu sama lain setiap kali pasien menjadwalkan janji temu. Hubungan juga punya beragam jenis, yang mendokumentasikan bagaimana sebuah *instance* dari suatu objek dapat dikaitkan dengan *instance* lainnya. Angka ditempatkan pada jalur asosiasi untuk menunjukkan kejadian minimum dan maksimum yang dapat dihubungkan melalui asosiasi dalam format angka minimum.. angka maksimum



Gambar 2.10 Contoh penggunaan notasi asosiasi  
(Dennis, Wixom, & Tegarden, 2015)



Gambar 2.11 Contoh penggunaan notasi asosiasi dengan kardinalitas  
(Dennis, Wixom, & Tegarden, 2015)

Pada gambar 2.11, terdapat tiga contoh asosiasi tambahan yang digambarkan, Faktur berhubungan dengan Invoice (dan sebaliknya), Pilot berhubungan dengan Pesawat, dan Ban Cadangan berhubungan dengan Bagasi. Dari gambar 2.11 tersebut menjelaskan bahwa sebuah *instance*

dari kelas Faktur harus diasosiasikan dengan satu *instance* dari kelas Invoice dan bahwa sebuah *instance* dari kelas Invoice dapat diasosiasikan dengan nol atau lebih *instance* dari kelas Faktur, sehingga sebuah *instance* dari kelas Invoice. Kelas Pilot Menerbangkan nol atau lebih *instance* dari kelas Pesawat, dan sebuah *instance* dari kelas Pesawat dapat diterbangkan oleh nol atau lebih *instance* dari kelas Pilot. Terakhir, kita melihat bahwa sebuah *instance* dari kelas Ban Serep adalah nol atau satu *instance* dari kelas Bagasi, sedangkan sebuah *instance* dari kelas Bagasi dapat berisi nol atau satu *instance* dari kelas Ban Serep.

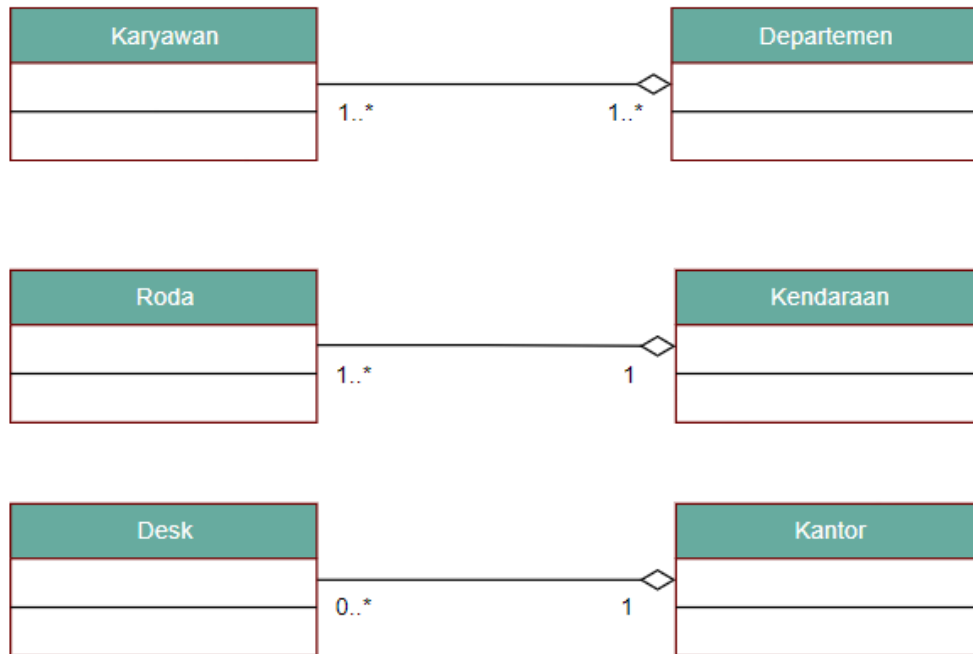
### **Generalisasi**

Generalisasi menunjukkan bahwa satu kelas (sub kelas) mewarisi dari kelas lain (super kelas), artinya properti dan operasi super kelas juga berlaku untuk objek sub kelas tersebut. Jalur generalisasi ditunjukkan dengan garis padat dari subkelas ke super kelas dan panah berongga yang menunjuk ke super kelas dapat dilihat pada gambar 2.10. yang menunjukkan bahwa dokter, perawat, dan resepsionis adalah semua sub kelas dari kelas karyawan dan karyawan serta pasien tersebut adalah sub kelas dari kelas partisipan.

### **Agregasi**

Agregasi digunakan untuk menggambarkan hubungan logis bagian-bagian dan digambarkan pada *class* diagram UML dengan Simbol berlian berongga atau putih. Sebagai contoh dapat dilihat pada gambar 2.12. Pada gambar tersebut ada tiga logika agregasi ditampilkan. Logika menyiratkan bahwa suatu bagian dapat diasosiasikan dengan beberapa keseluruhan atau relatif mudah jika bagian tersebut dihilangkan dari keseluruhan. Misalnya, sebuah *instance* dari kelas Karyawan bagian dari sebuah *instance* dari setidaknya satu *instance* dari kelas Department, sebuah *instance* dari kelas Roda bagian dari sebuah *instance* dari kelas Kendaraan, dan sebuah *instance* dari kelas Desk bagian dari sebuah *instance* dari kelas Kantor. Jelasnya, dalam banyak kasus, seorang

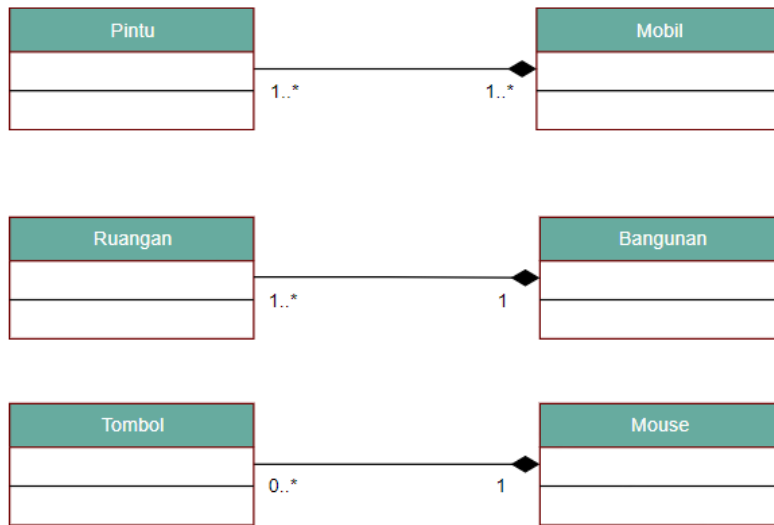
karyawan dapat dikaitkan dengan lebih dari satu departemen, dan relatif mudah untuk melepas roda dari kendaraan atau memindahkan desk dari kantor



Gambar 2.12 Contoh penggunaan notasi agregasi  
(Dennis, Wixom, & Tegarden, 2015)

### Composition (Komposisi)

*Composition* digunakan untuk menggambarkan bagian fisik dari hubungan dan disimbolkan dengan berlian hitam. Fisik menyiratkan bahwa suatu bagian hanya dapat diasosiasikan dengan satu kesatuan. Sebagai contoh pada gambar 2.13, tiga *composition* fisik diilustrasikan: sebuah contoh sebuah pintu dapat menjadi bagian dari sebuah mobil saja, sebuah contoh sebuah ruangan dapat menjadi bagian dari sebuah contoh dari sebuah bangunan saja, dan sebuah instance dari sebuah tombol hanya dapat menjadi bagian dari satu mouse.



Gambar 2.13 Contoh penggunaan notasi komposisi  
(Dennis, Wixom, & Tegarden, 2015)


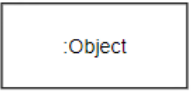
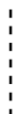
Terkadang, dalam banyak kasus, perbedaan yang dapat dicapai dengan memasukkan agregasi (berlian putih) dan komposisi (berlian hitam) dalam diagram kelas mungkin tidak sepadan dengan penambahan notasi grafis tambahan untuk dipelajari klien. Oleh karena itu, banyak ahli UML memandang penyertaan notasi agregasi dan komposisi ke diagram kelas UML hanya sebagai “pemanis sintaksis” dan tidak diperlukan karena informasi yang sama selalu dapat digambarkan hanya dengan menggunakan sintaksis asosiasi (Dennis, Wixom, & Tegarden, 2015, hal. 183).


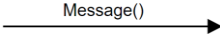
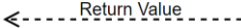
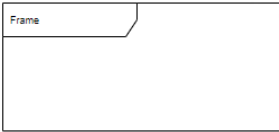
#### e. *Sequence* Diagram

Dalam pendekatan berorientasi berbasis objek, aliran informasi dicapai melalui pengiriman pesan dari aktor atau bolak-balik dari objek internal. *Sequence* diagram digunakan untuk menggambarkan aliran informasi masuk dan keluar dari bagian sistem. *Sequence* diagram mendokumentasikan masukan dan keluaran serta mengidentifikasi interaksi antara aktor dan sistem. *Sequence* diagram merupakan alat untuk membantu desain awal antar muka pengguna dengan mengidentifikasi informasi spesifik yang mengalir dari pengguna ke dalam sistem dan informasi yang mengalir keluar dari sistem kembali ke pengguna (Satzinger, Jackson, & Burd, 2016). Manfaat utama dari diagram urutan

adalah kemampuan untuk menata pesan dari atas ke bawah untuk menekankan urutan pengaktifan. Pada tabel 2.6 merupakan daftar sintaksis dan notasi dalam *sequence diagram*.

*Tabel 2.6 sintaksis dan notasi sequence diagram*

Sintaksis/Notasi	Nama notasi	Keterangan
	<i>Actor</i>	<ul style="list-style-type: none"> <li>• Mewakili seseorang (atau peran) yang berinteraksi dengan sistem.</li> <li>• Berpartisipasi secara berurutan dengan mengirim dan/atau menerima pesan</li> <li>• Digambarkan dengan simbol figur tongkat</li> </ul>
	<i>Object</i>	<ul style="list-style-type: none"> <li>• Berpartisipasi secara berurutan dengan mengirim dan/atau menerima pesan.</li> <li>• Digambarkan dengan simbol persegi panjang dengan nama objek</li> <li>• Titik dua sebelum nama kelas yang digarisbawahi adalah bagian notasi objek yang sering digunakan namun opsional untuk menunjukkan bahwa objek tersebut adalah objek kelas yang tidak disebutkan namanya.</li> </ul>
	<i>Object Lifeline</i>	<ul style="list-style-type: none"> <li>• Menunjukkan urutan pesan dari atas ke bawah</li> </ul>

		<ul style="list-style-type: none"> <li>• Digambarkan dengan garis putus-putus dengan arah vertikal</li> </ul>
	<i>Activity Lifeline</i>	<ul style="list-style-type: none"> <li>• Menunjukkan saat suatu objek mengirim atau menerima pesan</li> <li>• Digambarkan dengan simbol persegi panjang sempit yang ditempatkan di atas garis hidup.</li> </ul>
	<i>Message</i>	<ul style="list-style-type: none"> <li>• Menyampaikan informasi dari satu objek ke objek lainnya.</li> <li>• Panggilan operasi diberi label dengan pesan yang dikirim dan panah padat</li> </ul>
	<i>Return</i>	<ul style="list-style-type: none"> <li>• Mengembalikan nilai dari satu objek ke objek lainnya diberi</li> <li>• label dengan nilai yang Dikembalikan dan ditampilkan sebagai panah putus-putus.</li> </ul>
	<i>Frame</i>	<ul style="list-style-type: none"> <li>• Menunjukkan konteks diagram urutan</li> <li>• <i>Frame loop</i> menunjukkan sekumpulan pesan dalam satu <i>loop</i>.</li> <li>• <i>Frame Alternatif</i> merupakan logika jika-maka-lainnya</li> </ul>

---

pernyataan atau mengalihkan pernyataan, yang memungkinkan pengaktifan kumpulan pesan yang berbeda.

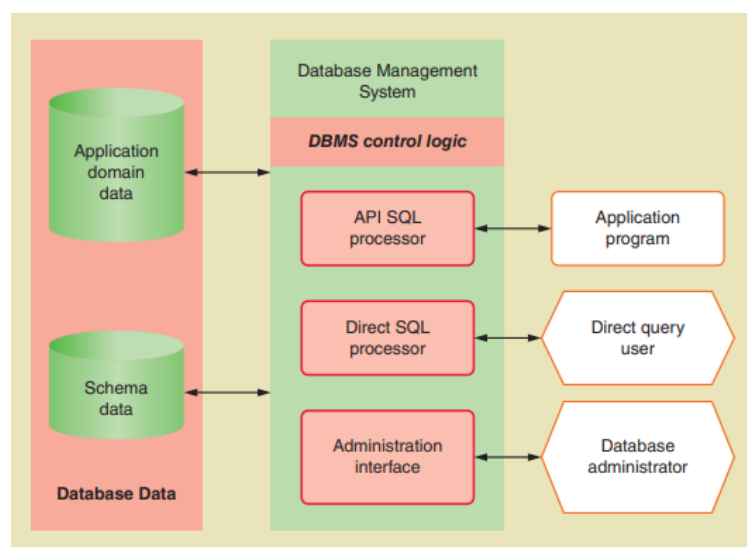
- *Frame Opt* adalah pemanggilan opsional dari sekumpulan pesan.

---

*Sumber :* (Satzinger, Jackson, & Burd, 2016)

#### 2.2.4. *Databases dan Database Management System (DBMS)*

*Database* merupakan kumpulan data yang terintegrasi yang disimpan, dikelola, dan diatur secara terpusat. Biasanya, basis data menyimpan informasi mengenai puluhan hingga ratusan kelas atau entitas. Pengelolaan dan pengendalian basis data dilakukan melalui sistem *Database Management System (DBMS)*. DBMS adalah bagian perangkat lunak dari sistem yang umumnya diperoleh dan diinstal secara terpisah dari komponen perangkat lunak sistem lainnya, seperti sistem operasi (Satzinger, Jackson, & Burd, 2016). Contoh dari sistem manajemen basis data modern meliputi Microsoft SQL Server<sup>TM</sup>, Oracle, dan MySQL.



*Gambar 2.14 Komponen database dan DBMS dengan aktor yang berinteraksi*  
*Sumber :* (Satzinger, Jackson, & Burd, 2016)



Gambar 2.14 mengilustrasikan komponen *database* pada umumnya. Biasanya ketika mengacu pada data *database*, yang ditunjukkan di sebelah kiri, dan DBMS, yang ditunjukkan di tengah. Data *database* terdiri dari dua penyimpanan informasi terkait, data domain aplikasi, yang merupakan data aktual untuk domain masalah, dan skema, yang berisi informasi deskriptif tentang data domain aplikasi. Alasan mengapa menyertakan semua komponen ini ketika membahas *database* adalah, karena data domain aplikasi tidak dapat diakses tanpa DBMS dan data skema.

Data skema atau skema data diperlukan untuk merinci struktur dan aturan yang mengatur cara mengakses domain aplikasi data. Skema sering disebut sebagai meta data atau informasi tentang data itu sendiri yang mencakup beberapa hal seperti:

- Deskripsi informasi mengenai data yang disimpan dalam penyimpanan data fisik.
- Komponen *database* yang mencakup pengorganisasian setiap item data dalam kelompok tingkat tinggi, seperti tabel.
- Hubungan antar tabel, seperti penunjuk dari objek pelanggan ke objek penjualan terkait.
- Rincian item data individual seperti tipe, panjang, lokasi, indeks item data, serta kontrol akses dan konten yang mencakup nilai yang diperbolehkan untuk setiap item data, ketergantungan nilai di antara beberapa item data, dan daftar pengguna yang memiliki izin untuk membaca atau memperbarui item data.

DBMS memiliki empat komponen utama yaitu *Application Programming Interface* (API), antarmuka SQL atau kueri langsung, antarmuka administratif, serta serangkaian program dan sub rutin akses data yang mendasarinya. Ilustrasi pada gambar 2.14 penggunaan SQL (Structured Query Language) sebagai bahasa kueri untuk mengakses database. Pengguna, program aplikasi, dan administrator tidak langsung

mengakses data domain. Mereka berinteraksi dengan antarmuka DBMS dan memberitahu jenis data apa yang mereka butuhkan, menggunakan nama-nama yang terdefinisi dalam skema.

DBMS menggunakan skema untuk memverifikasi keberadaan data yang diminta dan hak akses pengguna yang bersangkutan. Jika permintaan tersebut valid, DBMS akan mengambil informasi terkait objek data dari skema dan menggunakan informasi ini untuk mengakses data domain atas nama program atau pengguna yang membutuhkan data tersebut. *Database* dan DBMS menyediakan sejumlah kemampuan yang krusial dalam mengakses dan mengelola data, termasuk hal-hal berikut:

- Kemampuan untuk banyak pengguna dan program aplikasi mengakses data secara bersamaan.
- Akses ke data tanpa memerlukan penulisan program aplikasi, dapat dilakukan melalui proses langsung menggunakan SQL.
- Penyediaan kontrol akses dan aturan konten yang konsisten dan beragam.
- Integrasi data yang tersimpan di berbagai server yang tersebar di lokasi-lokasi berbeda.

#### **2.2.5 Relational Database Management System**

*Relational Database Management System* (RDBMS) adalah jenis DBMS yang mengatur penyimpanan data dalam struktur tabel atau hubungan. Tabel dalam *database* relasional menyerupai tabel konvensional, berbentuk struktur data dua dimensi yang terdiri dari kolom dan baris. Meski demikian, istilah yang digunakan dalam *database* relasional sedikit berbeda dari istilah yang biasa digunakan dalam tabel konvensional (Satzinger, Jackson, & Burd, 2016). Satu baris dalam tabel relasional disebut sebagai baris, tupel, atau catatan, sementara kolom dalam tabel disebut sebagai atribut atau *field*. Setiap sel dalam tabel disebut sebagai nilai atribut, nilai *field*, atau elemen data. Seperti yang telah disebutkan, akses dan pengeditan data dalam *database* relasional dilakukan melalui bahasa *query* umum yang dikenal sebagai *Structured*



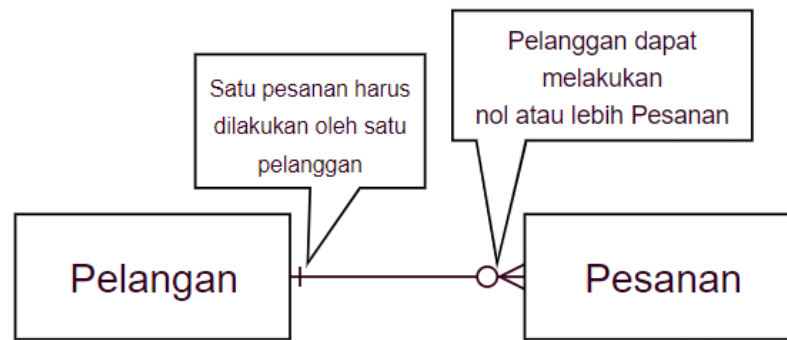
Pada gambar 2.6 tersebut terdapat dua tabel yang saling berhubungan tabel properti dan tabel kategori\_properti, tabel properti memiliki atribut *Foreign Key* yaitu id\_kategori\_properti yang menjadi kunci rujukan ke atribut *Primary Key* tabel kategori\_properti yaitu atribut id.

### 2.2.6 Entity Relationship Diagram (ERD)

Dalam pengembangan sistem sangat menekankan pada keperluan data untuk sistem baru dan menggunakan istilah "entitas data" untuk menggambarkan hal-hal yang dibutuhkan sistem untuk menyimpan informasi. Persyaratan data ini mencakup entitas data itu sendiri, atribut-atribut yang terkait, dan keterkaitannya (disebut sebagai "asosiasi" dalam UML) antara entitas data tersebut. Model yang sering digunakan oleh analis basis data adalah *Entity Relationship Diagram* (ERD). ERD bukanlah diagram UML, meskipun sering digunakan dan memiliki kemiripan yang tinggi dengan diagram kelas pada model domain UML (Satzinger, Jackson, & Burd, 2016).

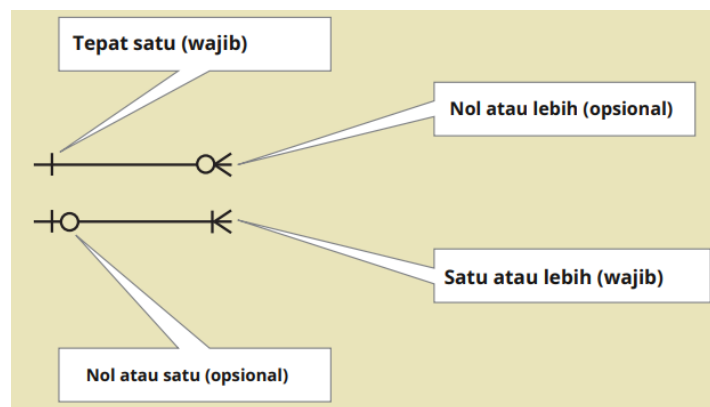
#### Contoh Notasi ERD

Pada diagram hubungan entitas, persegi panjang mewakili entitas data, dan garis yang menghubungkan persegi panjang menunjukkan hubungan antar entitas data. Gambar 2.17 menunjukkan contoh diagram hubungan entitas yang disederhanakan dengan dua entitas data, Pelanggan dan Pesanan. Setiap Pelanggan dapat melakukan banyak Pesanan, dan setiap Pesanan dilakukan oleh satu Pelanggan. Kardinalitasnya adalah satu-ke-banyak satu arah dan satu-ke-satu ke arah yang lain. Simbol kaki gagak pada garis di sebelah entitas data Pesanan menunjukkan banyak pesanan.

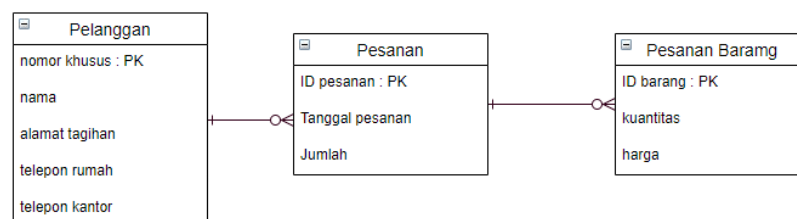


Gambar 2.17 Contoh ERD Sederhana  
Sumber : (Satzinger, Jackson, & Burd, 2016)

Namun simbol lain pada garis hubungan juga mewakili batasan kardinalitas minimum dan maksimum. Pada gambar 2.18 merupakan penjelasan simbol hubungan ERD.



Gambar 2.18 Simbol kardinalitas hubungan ERD  
Sumber : (Satzinger, Jackson, & Burd, 2016)

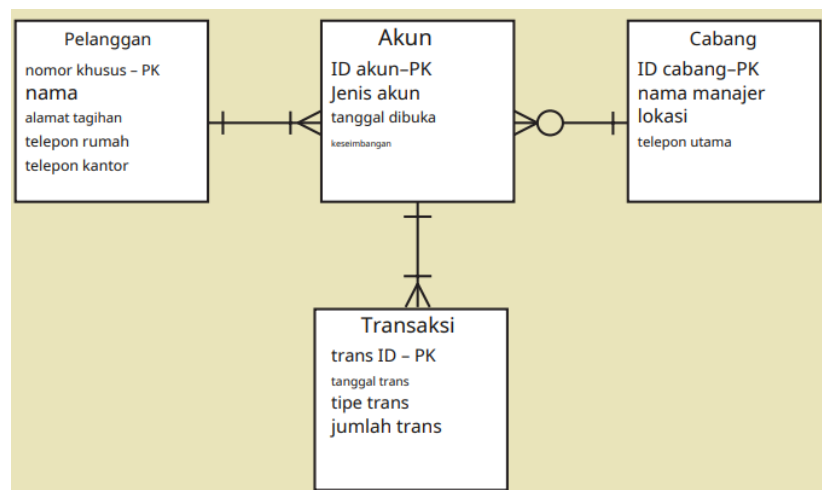


Gambar 2.19 Contoh ERD dengan atribut  
Sumber : (Satzinger, Jackson, & Burd, 2016)

Gambar 2.19 menunjukkan model yang diperluas untuk menyertakan item pesanan (satu atau lebih item spesifik yang disertakan dalam pesanan). Setiap pesanan berisi minimal satu item dan maksimal

banyak item (tidak mungkin ada pesanan jika tidak berisi minimal satu item). Misalnya, suatu pesanan mungkin mencakup kemeja, sepasang sepatu, dan ikat pinggang, dan masing-masing item ini dikaitkan dengan pesanan tersebut. Contoh ini juga memperlihatkan beberapa atribut setiap entitas data, Pelanggan memiliki nomor pelanggan, nama, alamat penagihan, dan beberapa nomor telepon. Setiap pesanan memiliki ID pesanan, tanggal pemesanan, dan sebagainya. Setiap item pesanan memiliki ID barang, kuantitas, dan harga.

Contoh lain ditunjukkan pada gambar 2.20. ERD ini diperuntukkan bagi bank yang mempunyai banyak cabang. Setiap cabang memiliki satu atau lebih akun. Setiap akun dimiliki oleh satu pelanggan dan menghasilkan satu atau lebih transaksi. Ada beberapa masalah lain yang perlu dipertimbangkan dalam contoh bank. Pertama, tidak ada entitas data bernama Bank. Hal ini dikarenakan ERD menunjukkan kebutuhan penyimpanan data pada bank. Hanya ada satu bank. Oleh karena itu, Bank tidak perlu dimasukkan ke dalam model. Hal tersebut adalah aturan umum yang berlaku untuk ERD. Jika sistem ini diperuntukkan bagi regulator bank negara, maka Bank akan menjadi entitas data yang penting karena terdapat banyak bank yang berada di bawah yurisdiksi regulator negara.



*Gambar 2.20 Contoh ERD dengan atribut*  
Sumber : (Satzinger, Jackson, & Burd, 2016)

### 2.3 Tinjauan Studi

Dalam penulisan penelitian ini dilakukan perbandingan dengan beberapa penelitian sebelumnya antara lain :

1. Tinjauan studi pertama diambil dari penelitian yang berjudul **“Sistem Informasi Pemasaran Dan Kredit Rumah Menggunakan Metode RAD”** yang ditulis oleh Mita Maulita, Marina Elsera, dan , Fachrul Rozi Lubis pada jurnal *Journal of Information Technology Research* Vol. 3, No. 1 Juli Tahun 2022. Penelitian ini difokuskan pada perumahan Grand Mekar Sari Park yang berlokasi strategis. Metode RAD dipilih karena pendekatannya yang inkremental dan cocok untuk proyek dengan waktu pengerjaan pendek. Pada bab metode penelitian menjelaskan langkah-langkah RAD, seperti perencanaan syarat-syarat, workshop desain RAD, dan implementasi. Hasil penelitian memperlihatkan antarmuka sistem yang dikembangkan, termasuk halaman utama, pendaftaran, login, promosi, data booking, dan data pembelian. Berbagai tampilan halaman disajikan untuk pengguna (pembeli), admin, dan pemilik. Penelitian ini berhasil mengembangkan sistem yang membantu pemasaran dan promosi properti serta mempermudah pengelolaan data bagi berbagai pengguna, termasuk pembeli, admin, dan pemilik.
2. Tinjauan studi kedua diambil dari penelitian yang berjudul **“Aplikasi Pemasaran Perumahan Pt. Griya Abee Makmur Ragajaya Citayam Kabupaten Bogor Menggunakan Metode Rapid Application Development (Rad) Berbasis Web”** yang ditulis oleh Muhammad Risqi Vramasatya, NM Faizah, dan Widyat Nurcahyo pada jurnal *Jurnal Indonesia : Manajemen Informatika dan Komunikasi* Vol 3 No 2, Juli-Desember (2022). Penelitian ini membahas tentang pengembangan aplikasi pemasaran perumahan berbasis web menggunakan metode Rapid Application Development

(RAD). Penelitian ini dilakukan untuk mengatasi tantangan dalam pemasaran properti, mengingat persaingan bisnis di industri pengembang properti semakin ketat. Dalam tahap perancangan, Unified Modeling Language (UML) digunakan untuk menggambarkan proses dan interaksi dalam aplikasi, termasuk proses login admin, penambahan data rumah, dan galeri gambar. Aplikasi ini dirancang untuk membantu pengembang properti mempromosikan perumahan, memberikan informasi lengkap kepada calon pembeli, serta menyediakan sarana pemesanan yang mudah. Keunggulan dari aplikasi berbasis web ini adalah memungkinkan pengguna, termasuk admin dan calon pembeli, untuk dengan mudah mengakses informasi perumahan dan melakukan pemesanan dengan efisien. Penggunaan teknologi web dan metode RAD dalam pengembangan aplikasi bertujuan untuk mengatasi tantangan dalam mempercepat proses pemasaran perumahan. Hasil dan pembahasan jurnal menguraikan berbagai diagram yang menggambarkan proses dan interaksi dalam aplikasi, seperti proses login admin, penambahan data rumah, dan galeri gambar. Aplikasi ini diharapkan dapat meningkatkan efisiensi dan efektivitas pemasaran perumahan serta memberikan layanan yang lebih baik kepada calon pembeli.

3. Tinjauan studi ketiga diambil dari penelitian yang berjudul **“Sistem Informasi Pemasaran Properti Berbasis Website Pada FAV Multi Sarana Bekasi”**. Terbit pada Juni 2019 Vol 3 No 2 oleh *Bina Insani Ict Journal* ditulis oleh Hafiz Al Farisyi, dan , Endang Retnoningsih. Penelitian ini bertujuan untuk merancang sebuah aplikasi yang efisien dan efektif dalam hal pemasaran perumahan dengan menggunakan metode Rapid Application Development (RAD) dan berbasis web. Berdasarkan hasil penelitian pada FAV Multi Sarana, dihasilkan Sistem Informasi Pemasaran Properti Berbasis Website. Sistem ini membantu dalam mengatasi beberapa permasalahan yang dihadapi perusahaan, seperti sulitnya mencari informasi perusahaan, informasi terbatas



mengenai properti yang dijual, dan ketidakterintegrasiannya data properti dengan anggota. Sistem ini terdiri dari berbagai komponen seperti database MySQL, perangkat keras (hardware) pendukung, dan software pendukung seperti PHP dan Javascript. Sistem ini mencakup berbagai hal, termasuk halaman login, halaman properti, halaman tambah properti, halaman anggota, halaman penjualan, dan lain-lain.

4. Tinjauan studi keempat diambil dari penelitian yang berjudul **“Implementasi Sistem Informasi Pemasaran Rumah Dengan Pendekatan Sistem Berorientasi Objek Pada Developer Property”** yang ditulis oleh Mustar Aman, Riyanto, Suroso, Ipang Sasono, Yunianto Agung Nugroho pada Jurnal ICT : *Information Communication & Technology* Vol. 20, N0.1, Juli 2021, pp. 156-164. Penelitian ini bertujuan mengimplementasikan sistem informasi pemasaran berbasis web pada Developer Property. Pengembangan sistem diinisiasi oleh kebutuhan untuk mengatasi kendala dalam pemasaran properti yang belum terkomputerisasi. Peneliti mengadopsi metode deskriptif kualitatif dengan pengumpulan data melalui observasi, wawancara, dan studi pustaka. Dari analisis SWOT, ditemukan bahwa sistem pemasaran saat ini memiliki kekuatan berupa interaksi langsung dengan calon pembeli, namun masih lemah dalam aspek daya jangkauan dan pengawasan. Peluang terbuka lebar dengan penggunaan teknologi informasi dalam pemasaran, namun ancaman datang dari persaingan industri yang semakin ketat. Hasil dari penelitian ini adalah pengembangan perangkat lunak berbasis web yang mencakup informasi detail perumahan, harga, lokasi, dan pembayaran. Penelitian ini menggarisbawahi pentingnya perencanaan, pengorganisasian, pelaksanaan, dan pengawasan dalam sistem pemasaran. Untuk ke depannya, disarankan adanya peningkatan jumlah karyawan, peningkatan komunikasi, dan penggunaan teknologi untuk optimalisasi pemasaran properti.

5. Tinjauan studi kelima diambil dari penelitian yang berjudul **“Rancang Bangun Sistem Informasi Pemasaran Perumahan Berbasis Web Menggunakan View 360° Pada Perumahan P T. Anindya Inti Perkasa (Aip)”** yang ditulis oleh Alfredo Hidayah Tuwllah, Arfan Yunus, pada Jurnal Jurnal Ilmu Komputer Kharisma pada 30 September 2020. membahas secara mendalam mengenai implementasi sistem informasi pemasaran berbasis pandangan 360° untuk perumahan. Sistem ini diterapkan pada P.T. Anindya Inti Perkasa (AIP), sebuah pengembang perumahan, dengan tujuan menyediakan cara pemasaran alternatif dan dinamis. Studi ini merinci proses pembuatan aplikasi yang menggunakan pandangan 360° sebagai media pemasaran yang lebih realistis dan interaktif. Aplikasi ini memungkinkan calon pembeli melihat lokasi perumahan dan detail ruangan secara *online*, mengatasi keterbatasan kunjungan fisik terutama dalam situasi pandemi. Sistem ini dirancang dengan bahasa pemrograman PHP dan menggunakan basis data MySQL. Metode pengujian yang digunakan adalah Black Box, yang berfokus pada fungsionalitas dan respons aplikasi. Penelitian ini memberikan kontribusi dalam membantu perusahaan pengembang perumahan menghadapi tantangan pemasaran dengan solusi inovatif berbasis teknologi. Dengan menggabungkan pandangan 360° dan informasi spesifikasi perumahan, aplikasi ini mempermudah calon pembeli dalam memutuskan tanpa perlu mengunjungi lokasi fisik

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Objek Penelitian**

Objek penelitian ini adalah perusahaan MP Project yang bergerak dalam industri properti, khususnya sebagai pengembang perumahan.

##### **3.1.1 Sejarah Perusahaan**

MP Project, yang berdiri pada tahun 2017, adalah sebuah perusahaan induk yang fokus pada pengembangan proyek perumahan dan telah menancapkan posisinya sebagai salah satu pemain kunci di industri properti Indonesia. Perumahan Kenanga Town House di Kemang, Jakarta Selatan, adalah proyek pertama mereka yang sukses, menjadi tonggak penting bagi perjalanan perusahaan ini.

Seiring waktu, MP Project terus mengembangkan inovasi dan memperluas portofolio proyeknya. Salah satu proyek terbaru mereka adalah Stellar Jardin Residence di Bekasi Selatan, mencerminkan komitmen perusahaan untuk mengikuti tren dan memenuhi kebutuhan konsumen. Selain itu, mereka juga merintis proyek Alpen Hills di Pamulang, Tangerang Selatan. Proyek-proyek ini menjadi bagian penting dari tekad MP Project untuk memberikan properti berkualitas dengan kehadiran digital yang kuat. Dengan sejumlah proyek sukses, MP Project terus berkembang di tengah persaingan ketat di dunia properti.

##### **3.1.2 Visi dan Misi Perusahaan**

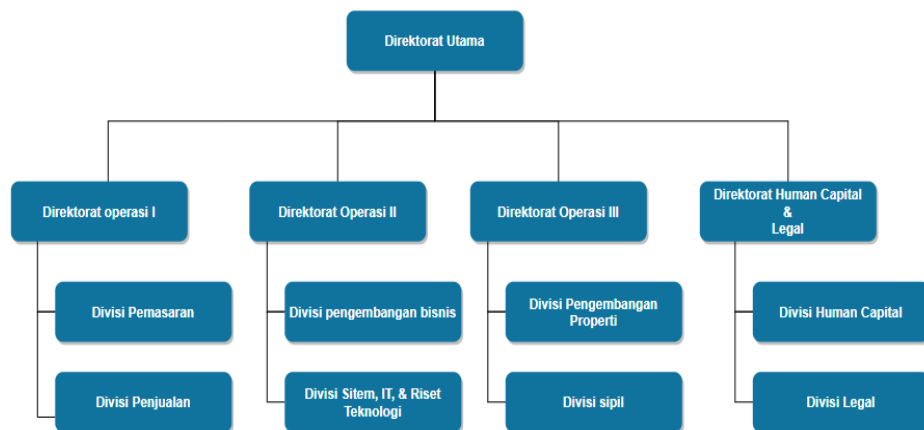
Visi MP Project adalah menjadi pionir dalam pengembangan properti yang bermutu tinggi dan berkelanjutan, menciptakan tempat tinggal yang nyaman dan mewah bagi masyarakat, serta menjadikan MP Project sebagai yang terdepan dalam industri properti.

Misi perusahaan ini meliputi:

1. Menghadirkan properti berkualitas tinggi dengan desain inovatif dan konstruksi yang kuat untuk memenuhi kebutuhan dan impian konsumen.

2. Menyajikan solusi perumahan ramah lingkungan dan berkelanjutan dengan penerapan teknologi terbaru dalam setiap proyek.
3. Menyediakan layanan pelanggan yang superior, responsif, dan profesional untuk memastikan kepuasan konsumen.
4. Berinovasi dalam pengembangan proyek perumahan dengan menciptakan konsep hunian yang unik dan menginspirasi.
5. Mengembangkan tim profesional yang berkomitmen, berkualitas, dan memiliki tingkat integritas yang tinggi untuk mencapai kesuksesan secara bersama-sama.

### 3.1.3 Struktur Organisasi



*Gambar 3.1 Struktur Organisasi MP Project*

berikut penjelasan lebih rinci mengenai struktur organisasi MP Project:

1. Direktorat Utama
  - Bertanggung jawab atas strategi global perusahaan, pengambilan keputusan utama, dan pengawasan keseluruhan operasi.
2. Direktorat Operasi I:
  - Divisi Pemasaran  
Melakukan riset pasar, perencanaan strategi pemasaran, serta pelaksanaan kampanye untuk meningkatkan kesadaran merek dan minat konsumen terhadap properti yang ditawarkan oleh perusahaan.
  - b. Divisi Penjualan

Menangani proses penjualan, mulai dari negosiasi kontrak hingga transaksi penjualan. Divisi ini juga bertanggung jawab terhadap hubungan langsung dengan calon pembeli dan pelayanan purna jual.

3. Direktorat Operasi II:

- Divisi Pengembangan Bisnis  
Bertugas menganalisis peluang pasar baru, merencanakan ekspansi bisnis, dan menjalin kemitraan strategis untuk pertumbuhan perusahaan.
- b. Divisi Sistem, IT & Riset Teknologi  
Memastikan infrastruktur teknologi informasi yang berkelanjutan, merancang dan mengimplementasikan sistem yang mendukung operasional perusahaan, serta melakukan riset untuk menerapkan teknologi terkini dalam proyek-proyek properti.

4. Direktorat Operasi III:

- Divisi Pengembangan Properti  
Menangani proses pengembangan proyek, mulai dari perencanaan, desain, hingga pelaksanaan konstruksi properti. Divisi ini bekerja sama dengan tim arsitek, insinyur, dan kontraktor untuk memastikan proyek berjalan sesuai rencana.
- Divisi Sipil  
Bertugas mengawasi aspek teknis konstruksi, termasuk pembangunan infrastruktur, manajemen konstruksi, dan pemantauan mutu proyek.

5. Direktorat Operasi IV:

- Divisi Human Capital  
Menangani rekrutmen, pengembangan, serta manajemen karyawan, termasuk program pelatihan, pengembangan karir, dan manajemen kinerja.
- Divisi Legal

Bertanggung jawab atas kepatuhan hukum perusahaan, menyusun dan meninjau kontrak, menangani perizinan, serta menangani segala aspek hukum dalam operasional perusahaan.

Struktur organisasi ini menunjukkan tanggung jawab masing-masing divisi di MP Project dalam mengelola aspek yang berbeda dari operasi perusahaan properti, mulai dari pengembangan proyek, pemasaran, penjualan, teknologi, hingga sumber daya manusia dan kepatuhan hukum.

### **3.2 Metode Penelitian**

Metode penelitian yang digunakan ini adalah metode R&D (*Research and Development*). Metode penelitian *Research and Development* merupakan suatu pendekatan yang banyak digunakan dalam berbagai bidang seperti pendidikan, psikologi, ilmu sosial, dan teknologi.

Menurut Muqdamien (2021) metode R&D merupakan pendekatan penelitian yang menghasilkan inovasi, baik dalam menciptakan produk baru maupun mengembangkan produk yang sudah ada agar lebih menarik dan sesuai dengan tujuan pembelajaran dari suatu pokok bahasan tertentu melalui proses pengumpulan data, melakukan analisis literatur, dan melakukan penelitian awal untuk memahami situasi atau masalah yang ada. Dari definisi di atas dapat dijelaskan bahwa metode *Research and Development* (R&D) merupakan metode yang digunakan untuk mengembangkan dan memperbaiki produk, proses, atau layanan dengan mengandalkan penelitian ilmiah.

#### **3.2.1 Tahap Pengumpulan Data**

Untuk memperoleh data yang diperlukan dalam pengembangan sistem yang akan dibuat, digunakan beberapa tahap pengumpulan data berikut ini:

1. Wawancara

Metode ini melibatkan kegiatan tanya jawab kepada pihak-pihak yang terkait pada proses penjualan. Melalui wawancara, bagian marketing, staf penjualan, dan pimpinan dapat dimintai pendapat mengenai proses penjualan serta kendala-kendala yang terjadi. Hal ini bertujuan untuk

memperoleh pemahaman langsung semua pihak yang terlibat dalam proses pemasaran properti perusahaan.

## 2. Analisis Dokumen

Metode ini melibatkan analisis terhadap dokumen dan data pendukung yang telah diberikan seperti dokumen Surat Pemesanan Rumah (SPR), dan Kuitansi Pemesanan. Tujuan dari analisis dokumen ini adalah untuk memahami struktur data yang digunakan dalam proses pemesanan properti.

## 2. Observasi

Melalui observasi, peneliti dapat mendapatkan pemahaman yang lebih mendalam tentang bagaimana proses penjualan properti berlangsung di dalam perusahaan. Hal ini mencakup pemahaman tentang bagaimana tim penjualan berinteraksi dengan calon konsumen, bagaimana data properti dikelola, serta identifikasi potensi kendala atau masalah yang mungkin muncul selama proses ini. observasi dilakukan dengan mengamati secara langsung aktivitas dan interaksi yang terjadi dalam perusahaan. Peneliti akan menghadiri pertemuan-pertemuan tim pemasaran, melihat bagaimana calon konsumen berinteraksi dengan tim penjualan, dan mencatat proses-proses yang terjadi dalam manajemen data properti.

### 3.2.1 Tahap Pengembangan Sistem

Rencana Pengembangan sistem akan mengikuti pendekatan *Rapid Application Development* (RAD). Yang terdiri beberapa tahap sebagai berikut :

#### 1. *Requirements Planning*

Tahap ini dimulai dengan pengumpulan dan analisis kebutuhan melalui interaksi langsung dengan tim marketing, bagian administrasi, serta pimpinan perusahaan. Metode seperti wawancara dan diskusi kelompok, akan digunakan untuk memahami secara mendalam proses pemasaran yang berlangsung dan hambatan yang dihadapi. Hasil dari

tahap ini akan membantu merumuskan kebutuhan awal yang menjadi dasar untuk perencanaan lebih lanjut.

## 2. *User Design*

Setelah memahami kebutuhan dari tahap sebelumnya, langkah selanjutnya adalah merancang dengan lebih terperinci bagaimana sistem informasi pemasaran properti akan dikembangkan. Pada fase ini, akan menggunakan Unified Modeling Language (UML) untuk merancang struktur sistem secara keseluruhan, termasuk komponen-komponen inti dan cara interaksi di antara mereka. Desain antarmuka pengguna/pengalaman pengguna (UI/UX) dan prototype juga akan dibuat agar mudah diakses dan memberikan informasi yang dibutuhkan oleh calon konsumen.

## 3. *Rapid Construction*

Tahap ini melibatkan pembuatan perangkat lunak sebenarnya, di mana coding atau pemrograman dimulai. Proses pengembangan perangkat lunak dilakukan secara iteratif dengan penerapan teknologi pemrograman berorientasi objek. Selama tahap ini, masukan dari pengguna terus dipertimbangkan untuk mencapai hasil yang sesuai dengan keinginan mereka. Proses ini akan diulang beberapa kali hingga mencapai versi akhir aplikasi, dengan tujuan memastikan kualitas dan kesesuaian sistem. Setelah berbagai komponen perangkat lunak selesai dibangun, integrasi dilakukan dan keseluruhan sistem diuji untuk memastikan kinerjanya yang optimal.

## 4. Implementasi

Tahapan ini merupakan langkah di mana sistem perangkat lunak yang telah disetujui sebelumnya diimplementasikan. Sebelum pelaksanaan, dilakukan pengujian menyeluruh untuk mengidentifikasi potensi kesalahan dalam sistem yang telah dibangun. Setelah pengujian, sistem dijalankan secara resmi dalam lingkungan produksi, diikuti oleh pelatihan pengguna. Selain itu, pemeliharaan sistem dilakukan untuk memastikan kinerja optimal dan pemenuhan kebutuhan pengguna tetap terjaga.



Dengan mengadopsi pendekatan RAD, penelitian ini diharapkan akan berhasil mengembangkan sistem informasi pemasaran properti yang responsif dan sesuai dengan kebutuhan yang berkembang, sehingga memberikan pengalaman yang lebih baik bagi calon konsumen dan membantu perusahaan dalam efisiensi operasional pemasaran.

### **3.3 Analisis Sistem yang Berjalan**

Dalam memperoleh hasil analisis pada sistem berjalan di perusahaan MP Project, maka dilakukan wawancara secara tatap muka terhadap narasumber yaitu Staf Penjualan, Marketing dan Pimpinan, yang hasilnya akan dirangkum kedalam sebuah tabel berisikan pertanyaan terkait perancangan sistem, yang sedang diteliti. Berikut adalah daftar pertanyaan wawancara yang dapat dilihat pada tabel 3.1

#### **3.3.1 Alur Sistem yang Berjalan**

Dalam analisis sistem yang berjalan, penelitian ini melibatkan pendekatan yang komprehensif dengan melakukan observasi dan wawancara terhadap *stackholder* operasional penjualan properti di MP Project untuk mengetahui alur sistem yang berjalan yaitu dengan pihak Marketing, Staf Penjualan dan Pimpinan.

Proses penjualan properti pada perusahaan MP Project terdapat beberapa prosedur yaitu sebagai berikut :

##### **1. Prosedur Survei Unit**

Prosedur Survei Unit dimulai dengan pelanggan yang mengunjungi kantor perusahaan dan bertemu dengan tim pemasaran. Mereka akan diberikan katalog dan daftar harga yang kemudian akan dijelaskan oleh tim pemasaran. Pelanggan akan dibimbing untuk memahami berbagai tipe unit hunian atau properti yang tersedia. Setelah itu, pelanggan akan melakukan survei unit bersama dengan tim pemasaran.

##### **2. Prosedur Pemesanan**

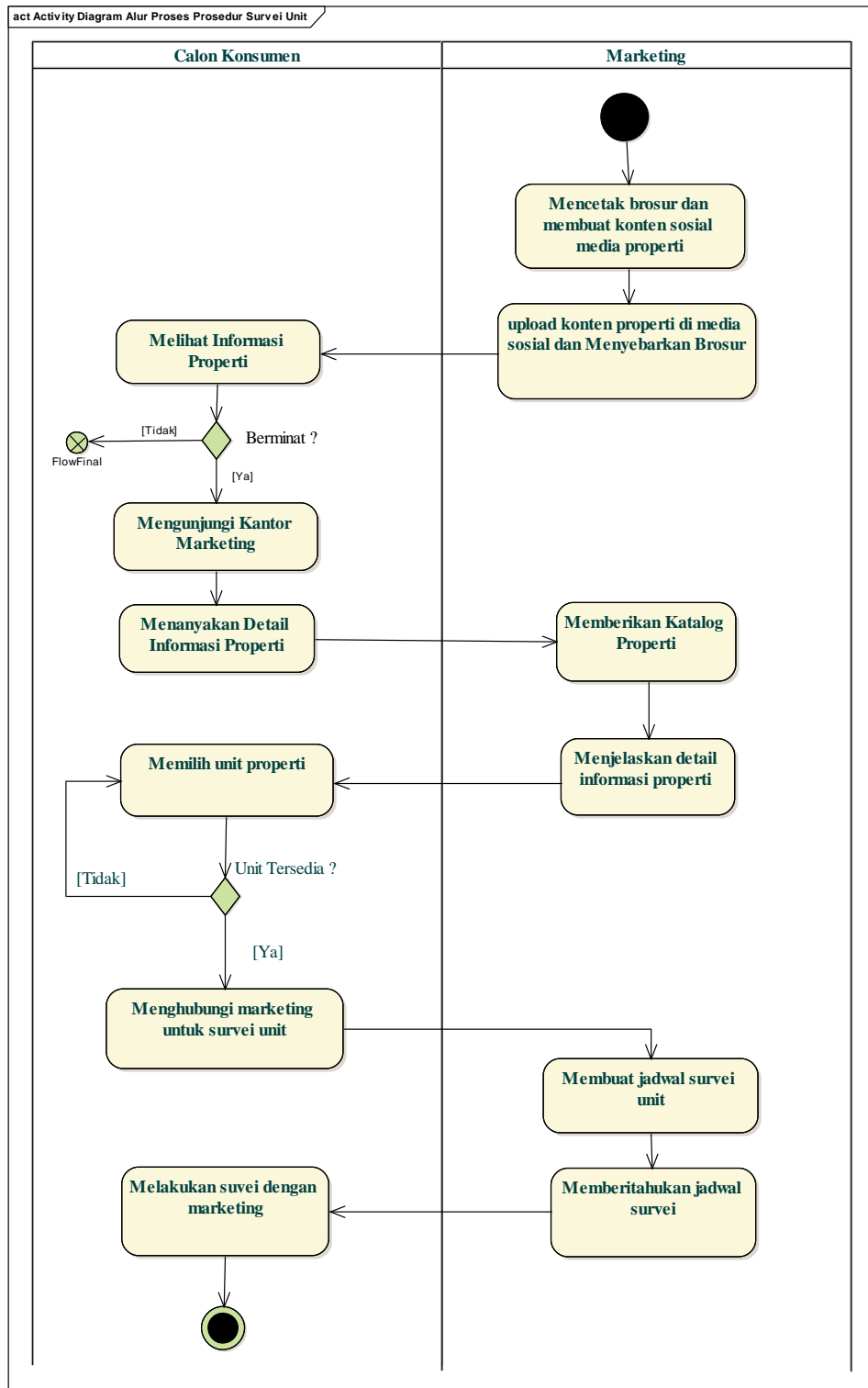
Jika konsumen tertarik dengan salah satu unit yang mereka survei, mereka dapat memutuskan untuk melakukan pemesanan. Pemesanan ini

biasanya dilakukan dengan cara memberikan konfirmasi kepada tim pemasaran bahwa mereka ingin membeli unit tertentu. Setelah konsumen melakukan pemesanan, mereka akan diminta untuk mengisi formulir pemesanan dan mengumpulkan berkas pembiayaan. Setelah semua berkas lengkap, konsumen diharuskan untuk membayar *booking fee*. *Booking fee* adalah sejumlah uang yang diberikan oleh konsumen sebagai tanda jadi atau untuk mengunci unit yang telah mereka pilih. Pembayaran ini dapat dilakukan dalam bentuk transfer bank atau secara langsung ke perusahaan. Setelah pembayaran *booking fee* diterima oleh perusahaan, konsumen akan diberikan Kuitansi sebagai bukti pembayaran. Kuitansi ini akan mencatat detail transaksi, termasuk jumlah *booking fee* yang dibayarkan, tanggal transaksi, dan unit yang dipesan setelah proses pemberkasan dan pembayaran selesai konsumen akan mendapatkan Surat Pemesanan Rumah (SPR) sebagai bukti pemesanan rumah atau unit properti.

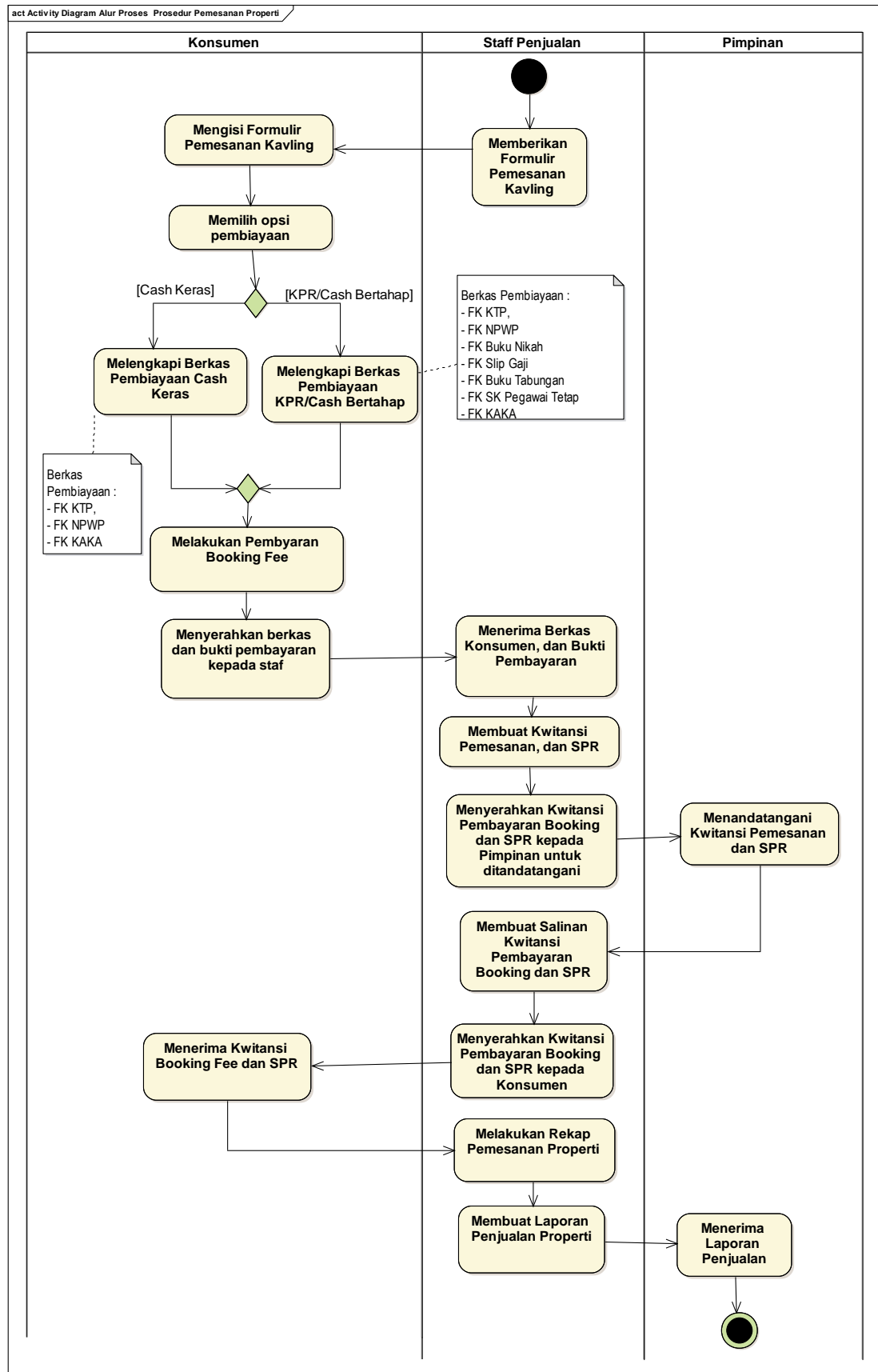
a. *Activity Diagram Alur Proses Survei Unit*

Proses survei unit di MP Project dimulai dari persiapan materi promosi seperti pencetakan brosur informatif dan konten promosi untuk media sosial oleh pihak marketing. Calon konsumen yang berminat dapat berkunjung ke kantor pemasaran MP Project untuk informasi lebih lanjut dan menerima katalog properti serta penjelasan rinci mengenai unit yang ada. Setelah memilih unit, konsumen memastikan ketersediaan unit yang diminati. Jika unit tersedia, konsumen menghubungi tim pemasaran untuk menjadwalkan survei. Namun, jika tidak tersedia, konsumen dapat memilih unit lain yang ada. Setelah jadwal survei dibuat, tim pemasaran memberitahukan kepada konsumen. Selanjutnya, konsumen bersama tim pemasaran melakukan survei unit, memungkinkan konsumen melihat langsung kondisi unit dan mendapatkan penjelasan detail.

Berikutnya pada Gambar 3.2 merupakan permodelan *activity diagram* dari alur proses survei unit.



Gambar 3.2 Activity Diagram Alur Proses Survei Unit



Gambar 3.3 Activity Diagram sistem berjalan

Pada gambar 3.3 Dalam proses pemesanan kavling, aktivitas dimulai saat staff penjualan memberikan formulir pemesanan kavling kepada konsumen. Konsumen kemudian mengisi formulir tersebut dan memilih jenis pembiayaan, baik Cash Keras atau KPR/Cash Bertahap. Jika konsumen memilih Cash Keras, mereka harus melengkapi dokumen seperti FK KTP, FK NPWP, dan FK KAKA. Sementara jika memilih KPR/Cash Bertahap, konsumen harus menyediakan dokumen tambahan seperti FK Buku Nikah, Slip Gaji, dan lainnya.

Setelah memilih pembiayaan dan melengkapi dokumen yang diperlukan, konsumen melakukan pembayaran booking fee dan menyerahkan berkas serta bukti pembayaran kepada staf penjualan. Staf penjualan menerima dokumen dan bukti pembayaran, kemudian membuat kwitansi pemesanan dan Surat Pemesanan Rumah (SPR) yang kemudian diserahkan kepada pimpinan untuk ditandatangani.

Setelah mendapatkan tanda tangan pimpinan, staf penjualan membuat salinan kwitansi dan SPR untuk diserahkan kepada konsumen. Konsumen menerima dokumen tersebut. Selanjutnya, staf penjualan melakukan rekap pemesanan properti dan membuat laporan penjualan yang akhirnya diserahkan kepada pimpinan untuk review.


Proses ini menunjukkan alur lengkap dari interaksi antara staff penjualan, konsumen, dan pimpinan, termasuk berbagai dokumen dan langkah-langkah yang terlibat dalam proses pemesanan properti.

### **3.3.2 Analisis Dokumen**

Berdasarkan dokumen dan data yang telah dikumpulkan dalam proses observasi ke lapangan, terdapat beberapa dokumen pendukung yang dapat dianalisis.

Bentuk dokumen dalam sistem penjualan properti pada MP Project sebagai berikut :

1. Dokumen Masukan :
  - a. Formulir Pemesanan Kavling



**MP PROJECT**  
 Jl. Cikunir Raya No.37, RT.001/RW.002, Jaka Mulya,  
 Kec. Bekasi Selatan, Kota Bekasi, Jawa Barat 17146  
 Telp. 081312344843

---

**FORMULIR PEMESANAN KAVLING**

Nomor : 006/MP/2023

Pada tanggal :  
 Nama Pemesan :  
 Tempat/tanggal lahir :  
 Nama dalam sertifikat/PPJB :  
 Alamat lengkap sesuai KTP :  
 :  
 Alamat surat-menyurat :  
 Nomor telepon rumah/kantor :  
 Nomor handphone :  
 Nomor KTP/Paspor/SIM :  
 Uang Sebesar ..... (.....)  
 guna membayar pesanan kavling untuk pembelian rumah di:  
 Nama Perumahan :  
 Nomor Kavling :  
 Tipe standar (lb/lt) :  
 Harga jual standar :  
 Terbilang :  
 Tipe kesepakatan (lb/lt) :  
 Harga jual kesepakatan :  
 Terbilang :

Bekasi, .....

Konsumen

.....

Marketing

.....

*Gambar 3.3 Formulir Pemesanan Kavling*

Nama Dokumen : Formulir Pemesanan Kavling

Fungsi : Untuk mengumpulkan informasi dan detail pemesanan kavling dari konsumen

Sumber : Staf Penjualan

Tujuan : Konsumen

Media : Kertas

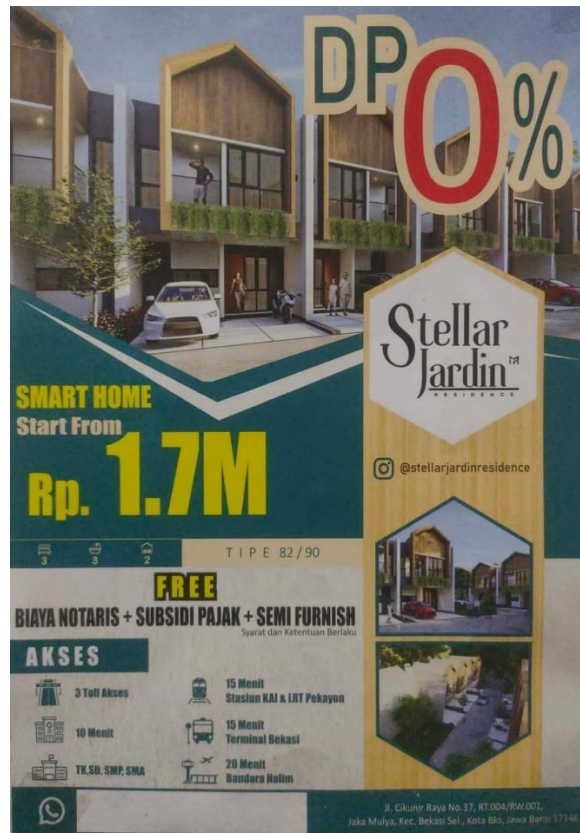
Jumlah : 1 Lembar

Frekuensi : Setiap kali ada permintaan pemesanan kavling

Bentuk : Gambar 3.3

2. Dokumen Keluaran :

a. Brosur Properti



Gambar 3.4 Brosur Properti

Nama Dokumen : Brosur Properti

Fungsi : sebagai materi promosi yang memberikan informasi terperinci tentang properti yang ditawarkan

Sumber : Marketing

Tujuan : Calon Konsumen

Media : Kertas

Jumlah : 1 Lembar

Frekuensi : Terjadi secara berkala tergantung pada penambahan properti baru atau strategi pemasaran yang direncanakan

Bentuk : Gambar 3.4

## b. Katalog Properti



Gambar 3.5 Katalog Properti

Nama Dokumen : Katalog Properti

Fungsi : Menyediakan informasi properti yang lebih lengkap

Sumber : Marketing

Tujuan : Calon Konsumen

Media : Kertas


Jumlah : 2 Lembar

Frekuensi : Terjadi secara berkala tergantung pada penambahan properti baru atau strategi pemasaran yang direncanakan

Bentuk : Gambar 3.5

## c. Kuitansi *Booking Fee*





**MP PROJECT**  
Jl. Cikunir Raya No.37, RT.001/RW.002, Jaka Mulya,  
Kec. Bekasi Selatan, Kota Bekasi, Jawa Barat 17146  
Telp. 081312344843

**KWITANSI BOOKING FEE**  
Nomor : 01 / MP/ 2023/2024

---

Nama Lengkap : .....

Alamat : .....

Telp / No. HP : .....

Komplek Perumahan : .....

Tipe Rumah : .....

Harga Rumah : .....

Sistem Pembayaran : .....

Jenis Setoran : ☒ Cash Keras ☐ Cash Bertahap

☒ KPR

Terbilang :

Untuk pembayaran :

Jumlah :

Bekasi,

\_\_\_\_\_  
PENYETOR

Soni Ahmad Muzaki  
\_\_\_\_\_  
DIREKTUR UTAMA

Perhatian :

1. Booking fee Rp. 5.000.000,- (lima juta rupiah)
2. Pembatalan sepihak dipotong 50% dari uang yang disetorkan ke developer
3. Pembayaran uang muka selambat – lambatnya 14 hari setelah booking fee/tanda jadi , bila tidak maka dianggap pembatalan sepihak.
4. Harga belum termasuk biaya: BPHTB, AJB, BALIK NAMA di NOTARIS dan biaya proses KPR.
5. Harga belum termasuk biaya lokasi rodut (Hook) atau tanah tidak standar.
6. Berkas selambat – lambatnya diserahkan 14 hari setelah booking fee.
7. Apabila konsumen terlambat menyerahkan berkas maka pihak developer berhak membatalkan sepihak atau mengalihkan kavlingnya ke calon konsumen lain.
8. Harga yang tercantum termasuk harga jual dan tidak termasuk biaya pengikatan muka.
9. Berkas tidak lolos Bank dipotong Rp. 1.000.000,- dari uang yang sudah dibayarkan.
10. Kwitansi ini bersifat sementara.
11. Jika sudah menerima kwitansi sementara, segera lakukan pembayaran di kasir dengan catatan menyerahkan kwitansi sementara beserta tanda jadi untuk mendapatkan kwitansi resmi sesuai dengan ketentuan perusahaan yang berlaku.
12. Pembayaran dianggap SAH apabila dilakukan di kantor marketing MP Project
13. Harga jual di atas dapat berubah sewaktu – waktu tanpa pemberitahuan terlebih dahulu dan menjadi hak sepenuhnya developer.

*Gambar 3.5 Kuitansi Booking Fee*

Nama Dokumen : *Kuitansi Booking Fee*

Fungsi : Sebagai bukti pembayaran *booking* unit

Sumber : Staf Penjualan

Tujuan : Konsumen


Media : Kertas

Jumlah : 1 Lembar

Frekuensi : Setiap setelah pembayaran *booking* unit

Bentuk : Gambar 3.5

d. Surat Pemesanan Rumah (SPR)



**MP PROJECT**  
Jl. Cikunir Raya No.37 RT.001/RW.002, Jaka Mulya,  
Kec. Bekasi Selatan, Kota Bekasi, Jawa Barat 17146  
Telp. 081312344843

**Surat Pemesanan Rumah**  
No. 05/MP /2023

---

**A. DATA PEMBELI**

Tanggal Pemesanan : \_\_\_\_\_

Nama : \_\_\_\_\_

Alamat : \_\_\_\_\_

No. KTP/ SIM : \_\_\_\_\_

Telp. : \_\_\_\_\_

HP. : \_\_\_\_\_

E-mail : \_\_\_\_\_

**B. DATA UNIT**

Type : \_\_\_\_\_

Unit : \_\_\_\_\_

Lokasi : \_\_\_\_\_

Kelebihan Tanah : \_\_\_\_\_

**C. HARGA JUAL**

Harga Price List : \_\_\_\_\_

Cash Back / Discount : \_\_\_\_\_

Harga Transaksi : \_\_\_\_\_

Catatan : \_\_\_\_\_

**D. CARA PEMBAYARAN**

Cash Keras : \_\_\_\_\_

Cash Bertahap : \_\_\_\_\_

KPR : \_\_\_\_\_

**E. RINCIAN PEMBAYARAN**

Uraian	Pembayaran	Tanggal	Keterangan
UTJ	Rp. _____		
DP RP.			
Biaya lain-lain Rp.			
Total DP All Rp.			
KPR			

Catatan :   
 : Klikan DP 1 dibayarkan 14 hari setelah UTJ  
 : Klikan DP 2 dibayarkan 30 hari setelah UTJ  
 : Klikan DP 3 s/d pelunasan dibayar setiap bulan berikutnya  
 bahwa konsumen telah membaca & menyatakan setuju dengan ketentuan dibawah ini:

**SYARAT & KETENTUAN :**

- > Apabila dalam waktu 14 (Empat Belas) hari sejak ditandatangani SPR ini, dokumen/ berkas untuk KPR belum diserahkan kepada kami dan Down Payment (DP) belum dibayarkan maka transaksi pembelian rumah ini secara otomatis dinyatakan batal dan Uang Tanda Jadi (UTJ) Tidak Dapat Dikembalikan, dan kami berhak menjual unit yang bersangkutan.
- > Jika KPR tidak disetujui oleh Bank atau Konsumen mengundurkan diri maka Down Payment (DP) dikembalikan 100% setelah unit terjual kepada Pihak Ketiga dan Uang Tanda Jadi (UTJ) tidak dapat dikembalikan.
- > Apabila plafon KPR yang disetujui Pihak Bank lebih kecil dari plafon yang diperhitungkan maka konsumen wajib menambah Down Payment (DP) dan harus diunasi sebelum akad kredit.
- > Pembayaran diklan Down Payment (DP) dianggap sah apabila dibayarkan melalui transfer dan atau dibayarkan kepada Pihak Developer diatas berdasar sll Developer.

**Pembayaran UTJ & DP hanya melalui:**  
Rek. Bank BCA  
**497-2023-888**  
a/n. MP PROJECT

**Kode Unik Konsumen**

\_\_\_\_\_

Yang Menyatakan,

Diketahui oleh,

Ditstujui oleh,

(.....)  
Konsumen

(.....)  
Marketing Head

(.....)  
Direktur Utama

*Gambar 3.6 Surat Pemesanan Rumah*

Nama Dokumen : Surat Pemesanan Rumah

Fungsi : Sebagai bukti pemesanan atau *booking*

Sumber : Admin

Tujuan : Konsumen

Media : Kertas

Jumlah : 1 Lembar

Frekuensi : Setiap setelah pembayaran *booking* unit

Bentuk : Gambar 3.6

### 3.4 Permasalahan

Berdasarkan hasil analisa terhadap proses penjualan yang ada di MP Project, melalui analisa alur sistem yang berjalan dan analisa dokumen, terdapat beberapa permasalahan pada sistem yang sedang berjalan saat ini, di antaranya adalah:

1. Penggunaan brosur fisik sebagai media promosi properti memiliki keterbatasan dalam memberikan informasi yang lengkap dan aktual. Terutama terkait biaya tambahan yang timbul saat terjadi kesalahan cetak, penambahan kuantitas, atau perubahan informasi. Setiap revisi pada brosur memerlukan biaya tambahan cetak dan desain yang bisa menjadi beban finansial yang cukup besar bagi perusahaan.
2. Sistem pengelolaan properti yang tidak dapat mengelola informasi properti secara *real-time*. Hal ini mengakibatkan kesulitan dalam akses informasi terkini, pembaruan, dan respons cepat terhadap perubahan serta kebutuhan pasar.
3. Dokumen-dokumen vital seperti KTP, NPWP, KK, dan persyaratan pembayaran lainnya tidak terorganisir dengan baik. Kurangnya pengorganisasian dan kelengkapan dalam manajemen dokumen konsumen memperlambat proses verifikasi dan penyelesaian transaksi penjualan properti.
4. Kurangnya sistem yang terpusat dan terstruktur untuk mengelola dokumen konsumen mengakibatkan kesulitan dalam melacak status, kelengkapan, dan keberadaan dokumen. Hal ini dapat menyebabkan keterlambatan dalam proses verifikasi dan transaksi penjualan properti.

### 3.5 Analisis Kebutuhan

Tahap analisis kebutuhan adalah langkah penting dalam pengembangan sistem informasi. Pada tahap ini peneliti mengidentifikasi dan mendefinisikan dengan jelas apa yang perlu dicapai oleh sistem yang akan dikembangkan. Tahap analisis kebutuhan ini melibatkan interaksi langsung dengan tim marketing, bagian staf penjualan, serta pimpinan perusahaan. Peneliti menggunakan metode *elicitation* (elisitasi) untuk mengumpulkan kebutuhan dari stakeholder. Hasil dari tahapan elisitasi ini dapat dilihat pada tabel 3.1 yang mencantumkan daftar

kebutuhan fungsional dan non-fungsional yang telah diidentifikasi selama proses analisis.

*Tabel 3.1 Hasil analisa kebutuhan fungsional dan non-fungsional*

<b>Fungsional</b>	
<b>Analisa Kebutuhan</b>	
<b>Saya ingin Sistem Dapat</b>	
1.	Menyediakan fitur registrasi akun untuk pengguna baru
2.	Menyediakan fitur pengelolaan data pribadi untuk konsumen
3.	Mampu melakukan autentikasi pengguna setiap akun yang terdaftar
4.	Menyediakan fitur pengelolaan informasi akun seperti pengantian password dan username/email untuk pengguna
5.	Memberikan hak akses halaman yang berbeda antara Konsumen, Marketing, Pimpinan, dan Staf Penjualan
6.	Mampu menyediakan informasi lengkap dan detail tentang setiap properti yang tersedia, termasuk deskripsi, foto, peta lokasi, harga, dan fasilitas.
7.	Sistem harus menyediakan kalkulator KPR yang memungkinkan calon konsumen untuk menghitung estimasi cicilan KPR berdasarkan suku bunga, tenor, dan jumlah pinjaman.
8.	Konsumen dapat memesan properti melalui proses reservasi, upload dokumen pembiayaan, pembayaran, dan konfirmasi pemesanan.
9.	Menyediakan informasi lengkap terkait riwayat pemesanan properti untuk konsumen
10.	Staf penjualan dapat mengelola produk, termasuk menambah produk properti, memperbarui informasi properti, dan menghapus properti
11.	Pimpinan dapat melihat informasi terkait laporan pemesanan
12.	Bagian marketing dapat melakukan pengelolaan dan mengkonfirmasi beserta memvalidasi data pemesanan
<b>Non Fungsional.</b>	
<b>Saya Ingin Sistem Dapat</b>	
1.	Antarmuka pengguna yang mudah digunakan dan dipahami
2.	Sistem harus dapat diakses di berbagai jenis web browser, termasuk Chrome, Firefox, Safari, dan Edge. Hal ini untuk memastikan bahwa calon konsumen dapat mengaksesnya tanpa batasan perangkat
3.	Sistem harus mampu memberikan respons cepat dan performa yang baik, terutama saat melakukan pencarian properti
4.	Sistem harus tersedia sepanjang waktu, dengan waktu henti minimal untuk pemeliharaan atau pembaruan rutin.