



UNIVERSITAS  
**AMIKOM**  
YOGYAKARTA

**2018**

# ANALISIS DAN PERANCANGAN SISTEM INFORMASI

*Program Studi D3 Manajemen Informatika  
Fakultas Ilmu Komputer  
Universitas AMIKOM Yogyakarta*

## **KATA PENGANTAR**

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa atas berkah dan karunia-Nya, sehingga Modul Analisis dan Perancangan Sistem Informasi ini dapat diselesaikan. Modul ini dibuat sebagai pedoman dalam melakukan kegiatan perkuliahan Analisis dan Perancangan Sistem Informasi yang merupakan salah satu matakuliah wajib pada Jurusan Diploma Tiga Manajemen Informatika Universitas AMIKOM Yogyakarta. Modul ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan kegiatan perkuliahan dengan lebih baik, terarah, dan terencana.

Pada setiap topik telah ditetapkan tujuan materi yang disampaikan dan semua tugas yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

## DAFTAR ISI

Kata Pengantar .....	i
Daftar Isi .....	ii
<b>BAB I. KERANGKA ANALISIS DAN DESAIN SISTEM .....</b>	<b>1</b>
1.1 Definisi Sistem .....	1
1.2 Klasifikasi Sistem Informasi .....	1
1.3 Tugas .....	5
<b>BAB II STAKEHOLDER SISTEM .....</b>	<b>6</b>
2.1 Stakeholder Sistem .....	6
2.2 Tugas .....	8
<b>BAB III PENGEMBANGAN SISTEM INFORMASI .....</b>	<b>9</b>
3.1 Definisi Pengembangan Sistem .....	9
3.2 Indikator Perlunya Pengembangan Sistem .....	9
3.3 Prinsip-prinsip Pengembangan Sistem .....	10
3.4 Tujuan Pengembangan Sistem .....	10
3.5 Siklus Hidup Pengembangan Sistem .....	11
<b>BAB IV ANALISIS SISTEM .....</b>	<b>13</b>
4.1 Aktifitas Pada Fase Analisis Sistem .....	13
4.2 Tugas .....	34
<b>BAB V PERANCANGAN MENGGUNAKAN DATA FLOW DIAGRAM (DFD) .....</b>	<b>35</b>
5.1 Pengertian DFD .....	35
5.2 Elemen dan Aturan Dalam Pembuatan DFD .....	35
5.3 Tingkatan Dalam Pembuatan DFD .....	39
5.4 Tugas .....	48
<b>BAB VI PERANCANGAN DATABASE MENGGUNAKAN     ENTITY RELATIONSHIP DIAGRAM (ERD) .....</b>	<b>49</b>
6.1 Definisi ERD .....	49
6.2 Komponen ERD .....	49
6.3 Tugas .....	57
<b>BAB VII PERANCANGAN SISTEM MENGGUNAKAN UNIFED MODELLING LANGUAGE (UML) 58</b>	
7.1 Konsep dasar OOP (Object Oriented Programming) .....	58
7.1.1 Object .....	58
7.1.2 Kelas (Class) .....	58
7.1.3 Enkapsulasi .....	59
7.1.4 Inheritance .....	59
7.1.5 Interface .....	61
7.2 Diagram Unifed Modeling Language (UML).....	61
7.3 Use Case Diagram (Diagram Use-case) .....	62
7.4 Activity Diagram .....	77
7.5 Class Diagram .....	83
7.6 Sequence Diagram .....	87
7.7 Tugas .....	93

BAB VIII PENGUJIAN BLACK BOX .....	94
8.1 Pengertian Black Box Testing .....	94
8.2 Tugas .....	96
DAFTAR PUSTAKA.....	97

## **BAB I**

### **KERANGKA ANALISIS DAN DESAIN SISTEM**

---

#### **Pertemuan ke-1**

**Tujuan : Mahasiswa memahami klasifikasi sistem informasi berdasarkan fungsi yang dimiliki**

---

#### **1.1 Definisi Sistem**

Definisi dari sistem itu sendiri adalah kumpulan elemen-elemen yg saling berhubungan atau terintegrasi dan mempunyai satu tujuan (Whitten, dkk (2004)).

Definisi sistem informasi menurut para ahli :

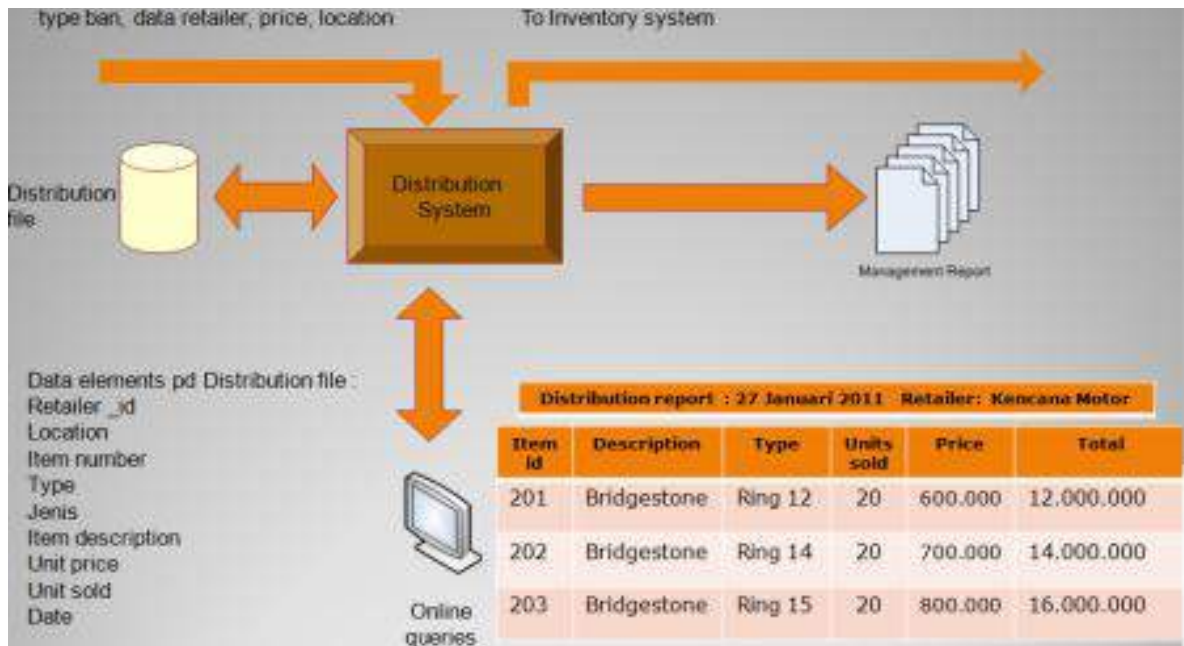
1. Menurut J.L. Whitten sistem informasi adalah pengaturan orang, data, proses, dan informasi teknologi/teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan, dan menyediakan sebagai output informasi yang diperlukan untuk mendukung organisasi (Whitten, dkk (2004)).
2. O'Brien mengatakan bahwa sistem informasi merupakan suatu kombinasi dari setiap unit yang dikelola oleh user atau manusia, hardware (perangkat keras komputer), software (perangkat lunak), jaringan komputer dan jaringan komunikasi data (komunikasi), dan juga database (basis data) yang mengumpulkan, mengubah, dan menyebarkan informasi tentang suatu organisasi. Jadi, pada dasarnya, sistem informasi memang harus memiliki elemen – elemen tersebut agar dapat berguna dan juga bekerja dengan optimal (O'Brien, dkk (2011)).
3. Leitch & Davis juga mengemukakan pendapat mereka mengenai sistem informasi. Sistem informasi adalah sebuah sistem yang terdapat di dalam suatu organisasi yang berfungsi untuk mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, kegiatan manajerial dan strategis dari suatu organisasi dan memberikan hasilnya dalam bentuk laporan bagi pihak – pihak luar (Leitch, Robert (1983)).
4. Davis (1991) mengatakan bahwa suatu sistem informasi adalah sebuah sistem yang menerima input data dan instruksi, mengolah data sesuai dengan instruksi dan mengeluarkan hasilnya. Dengan begitu, maka bisa disimpulkan bahwa suatu sistem informasi memiliki alur tertentu, mulai dari input hingga menjadi output yang bermanfaat (Davis and Olson (2012)).
5. Stair & Reynolds (2010) mengatakan bahwa sistem informasi merupakan suatu perangkat elemen atau komponen yang saling terkait satu sama lain, yang dapat mengumpulkan, mengolah, menyimpan dan juga menyebarkan data dan juga informasi, serta mampu untuk memberikan feedback untuk memenuhi tujuan suatu organisasi (Stair dan Reynolds (2017)).
6. Kenneth dan Laudon (2010) Laudon mengatakan bahwa yang dimaksud dengan sistem informasi adalah suatu komponen yang saling bekerja satu sama lain untuk mengumpulkan, mengolah, menyimpan dan juga menyebarkan informasi untuk mendukung kegiatan suatu organisasi, seperti pengambilan keputusan, koordinasi, pengendalian, analisis masalah, dan juga visualisasi dari organisasi (Laudon dan Laudon (2015)).

#### **1.2 Klasifikasi Sistem Informasi**

Sistem informasi berdasarkan fungsi yang dimiliki dapat diklasifikasikan menjadi (Whitten, dkk (2004)) :

1. Sistem pemrosesan transaksi / *Transaction processing system (TPS)* adalah sebuah sistem informasi yang mengcapture dan memproses data transaksi bisnis seperti mencatat data, memproses data dan menghasilkan informasi baku.

Contoh : Sistem informasi penjualan yang didalamnya berisi pencatatan transaksi penjualan dan pembelian di toko ban X. Semua data dan transaksi disimpan di dalam *database* untuk dilakukan pengolahan pada datanya sehingga bias menghasilkan suatu informasi atau laporan.

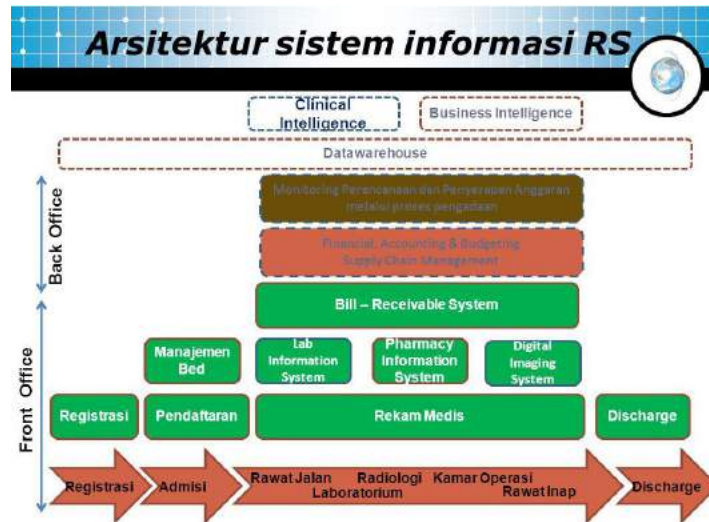


Gambar 1. Arsitektur sistem pemrosesan transaksi [1]

2. Sistem informasi manajemen / *Management Information System (MIS)* adalah sebuah sistem informasi yang menyediakan data untuk pelaporan berorientasi manajemen berdasarkan pemrosesan transaksi dan operasi organisasi (Whitten, dkk (2004)). Sedangkan Raymond McLeod Jr mendefinisikan Sistem Informasi Manajemen sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai yang mempunyai kebutuhan yang serupa. Informasi menjelaskan perusahaan atau salah satu sistem utamanya mengenai apa yang telah terjadi di masa lalu, apa yang sedang terjadi sekarang dan apa yang mungkin terjadi di masa depan (McLeod dan Schell (2006)).

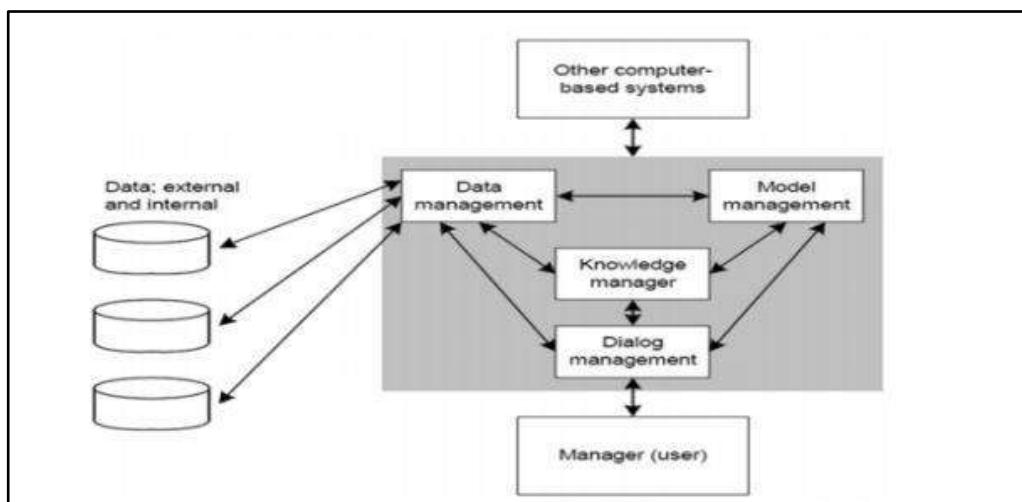
Sedangkan menurut Davis, Sistem Informasi Manajemen adalah sebuah sistem yang terintegrasi antara manusia dan mesin yang mampu memberikan informasi sedemikian rupa untuk menunjang jalannya operasi, jalannya manajemen dan fungsi pengambilan keputusan di dalam sebuah organisasi (Davis, G dan Olson, M., (2012)).

Contoh : Sistem Informasi Manajemen Rumah Sakit (SIMRS) dikembangkan untuk menangani Sistem Administrasi (*Bussiness Process*) dan Rekam Medis (*Medical Record*) di Rumah Sakit. Dirancang untuk dapat berjalan dengan baik pada semua Sistem Operasi (*Multi Platform Operating System*) seperti: Windows, Linux, Apple Machintosh, termasuk juga Tablet PC maupun Mobile Phone. Mengakomodasi Sistem Proses backup Automatic antar Server (*Redundant System*), Mengakomodasi Sistem Proses Pembagian Beban Kerja antar Server (*Paralel Computing*), Mengakomodasi Sistem Proses pooling data pada Sistem Layanan Awan (*Private Cloud System*) sehingga bisa diakses dimana saja, serta dukungan Komputasi bergerak dengan pemanfaatan media komunikasi layanan internet.



Gambar 2. Arsitektur sistem informasi manajemen rumah sakit

3. Sistem pendukung keputusan / *Decision Support System* adalah sebuah sistem informasi yang membantu dalam mengidentifikasi kesempatan pembuatan keputusan atau menyediakan informasi untuk membantu pembuatan keputusan (Whitten, dkk (2004)). Menurut Turban dan Aronson (2011), *Decision Support System* (DSS) atau Sistem Pendukung Keputusan (SPK) adalah sistem yang dimaksudkan untuk mendukung pembuat keputusan manajerial dalam situasi keputusan semiterstruktur dan terstruktur. SPK berfungsi sebagai tambahan atau pendukung bagi pembuat keputusan, dapat memperluas pengetahuan dan kemungkinan, namun tidak menggantikan penilaian. Sistem ini ditujukan untuk keputusan yang membutuhkan penilaian dan keputusan yang dapat diolah dengan algoritma atau secara teknis (Turban, dkk (2007)).  
Contoh : Sistem pendukung keputusan penerima dana bantuan RASKIN di Daerah Istimewa Yogyakarta.

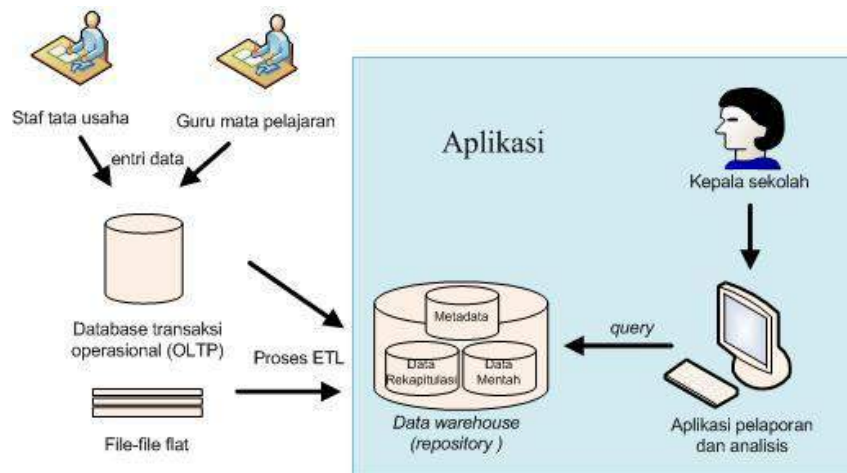


Gambar 3. Arsitektur sistem pendukung keputusan [8]

4. Sistem informasi eksekutif / *Executive Information System* adalah sebuah sistem informasi yang mendukung perencanaan dan penilaian kebutuhan manajer eksekutif (Whitten, dkk (2004)). Mc-Leod (2010) mendefinisikan Sistem Informasi Eksekutif (*Executive Information System* – EIS) merupakan suatu sistem yang menyediakan informasi bagi eksekutif mengenai kinerja keseluruhan perusahaan. Informasi dapat diambil dengan mudah dan dalam berbagai tingkat

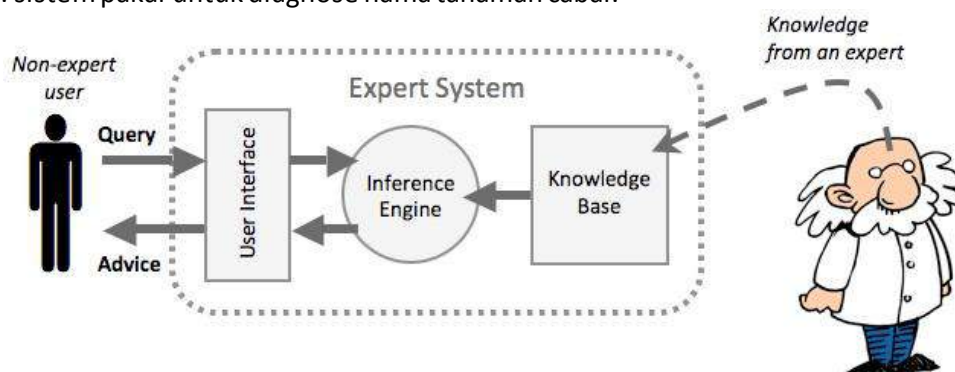
rincian (McLeod dan Schell (2006)). Indrajit (2000) mendefinisikan Sistem Informasi Eksekutif (*Executive Information System - EIS*) sebagai suatu alat (*tool*) yang dapat menghasilkan suatu sistem pelaporan yang tertinggi dalam suatu perusahaan. Yang dilakukan oleh perangkat EIS adalah melakukan penarikan data (*data extraction*) dan mensarikannya (*data summarizing*) dari suatu sumber data tertentu atau database yang ada dibawahnya.

Contoh : Contoh sistem informasi eksekutif (EIS) di lingkungan sekolah. Sistem informasi eksekutif yang dibangun merupakan software yang digunakan bagi para eksekutif (misalnya kepala sekolah, pengelola yayasan) dalam menjalankan roda organisasi. Aplikasi berisi modul yang menyediakan beragam informasi eksekutif yang dapat dimanfaatkan oleh eksekutif dalam memonitor perkembangan dan kualitas pengajaran, perkembangan prestasi siswa, kondisi keuangan organisasi hingga status perkembangan berbagai kegiatan pengajaran yang dilakukan di sekolah.



**Gambar 4.** Arsitektur Sistem Informasi Eksekutif

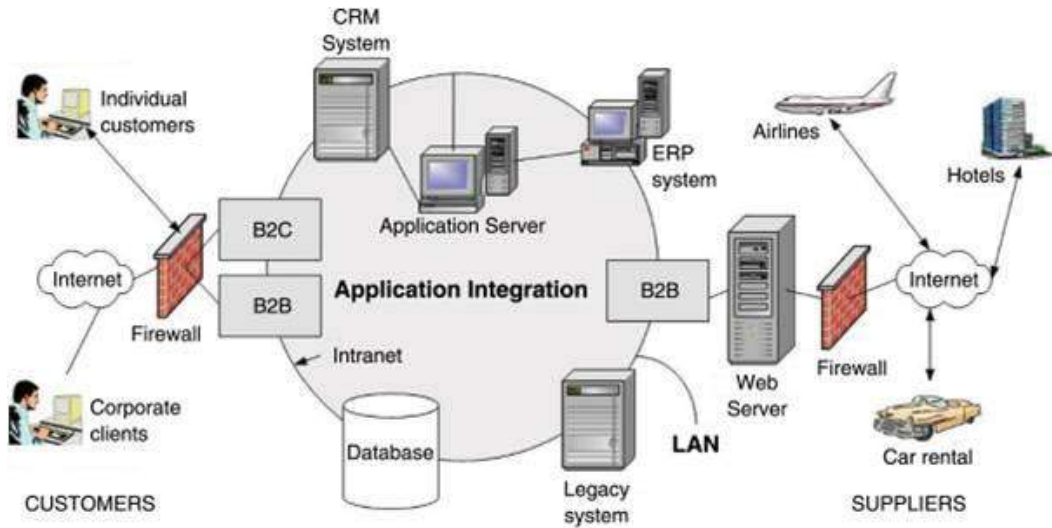
5. Sistem pakar (*Expert System*) adalah sebuah sistem informasi yang mengcapture keahlian seorang *expert* dan mensimulasikan keahlian tersebut untuk bisa digunakan oleh orang yang tidak ahli pada bidang tersebut (Whitten, dkk (2004)).  
Contoh : sistem pakar untuk diagnose hama tanaman cabai.



**Gambar 5.** Arsitektur Sistem pakar

6. Sistem komunikasi dan kolaborasi (*Communication and Collaboration System*) adalah sebuah sistem informasi yang dibuat untuk mengefektifkan komunikasi dan kemampuan kolaborasi antara pekerja, rekan kerja, pelanggan dan penyuplai (Whitten, dkk (2004)).  
Contoh : sistem pemesanan tiket pesawat, hotel di traveloka





**Gambar 6.** Gambaran arsitektur Sistem komunikasi dan kolaborasi

### 1.3 Tugas

1. Bentuklah kelompok dengan anggota sebanyak 4 orang !
2. Tentukan topik sistem informasi yang ingin anda buat sistemnya !

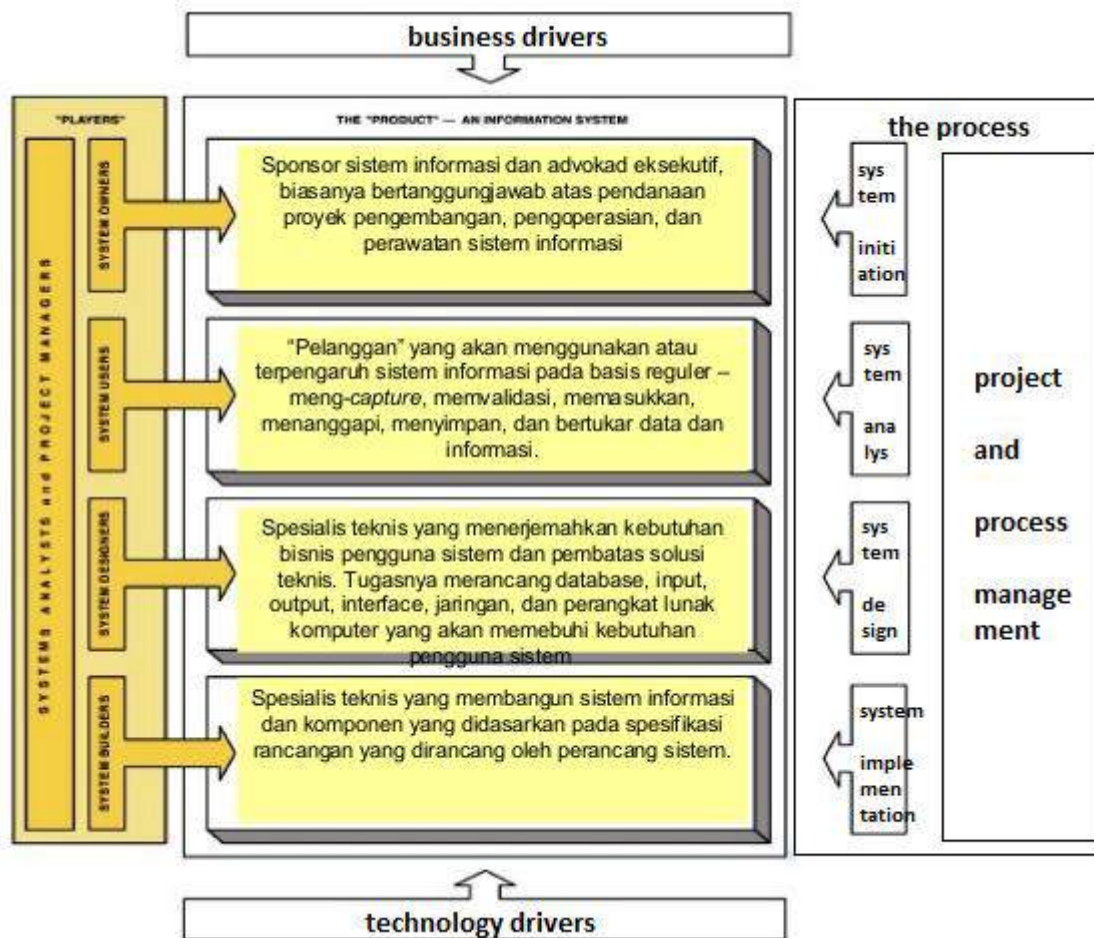
## BAB II STAKEHOLDER SISTEM

### Pertemuan ke-2

Tujuan : Mahasiswa memahami stakeholder yang terlibat dalam pengembangan dan pembangunan sistem informasi

#### 2.1 Stakeholder Sistem

Dalam pengembangan suatu sistem, harus didukung oleh personal-personal yang kompeten di bidangnya. Personal-personal ini disebut stakeholder. Stakeholder adalah orang yang memiliki ketertarikan pada sistem informasi yang sudah ada atau ditawarkan. Stakeholder bisa termasuk pekerja teknis dan nonteknis, bisa juga termasuk pekerja dalam dan luar (Whitten, dkk (2004)). Stakeholder yang terlibat dalam pengembangan dan pembangunan sistem informasi adalah : (lihat gambar 7)



Gambar 7. Stakeholder dalam pengembangan dan pembangunan sistem informasi [1]

Stakeholder menurut gambar 7 terdiri dari :

1. *System owners* (pemilik sistem)  
Pemilik sistem biasanya berasal dari tingkat manajemen dan cenderung tertarik dengan keuntungan. Keuntungan disini dapat didefinisikan sebagai :
  - a. Peningkatan keuntungan perusahaan
  - b. Pengurangan biaya bisnis
  - c. Biaya dan keuntungan sistem
  - d. Peningkatan pangsa pasar
  - e. Peningkatan efisiensi
  - f. Perbaikan pembuatan keputusan
  - g. Kesalahan lebih sedikit
  - h. Perbaikan keamanan
  - i. Kapasitas lebih besar
2. Pengguna sistem  
Pengguna sistem cenderung tidak mengacuhkan biaya dan keuntungan sistem, melainkan lebih memperhatikan fungsionalitas sistem yang disediakan untuk pekerjaan mereka dan kemudahan pemberlajaran dan penggunaan sistem.  
Contoh pengguna sistem internal organisasi:
  - a. Pekerja administrasi dan layanan
  - b. Staf teknis dan profesional (pekerja pengetahuan)
  - c. Supervisor, manajer menengah, dan manajer eksekutif.Contoh pengguna sistem eksternal organisasi:
  - a. Pelanggan
  - b. Pemasok
  - c. Rekan kerja atau partner
  - d. Karyawan yang bekerja di
    - rumah (*telecommute*)
    - bergerak (*mobile*), misal sales.
3. Desainer sistem  
Spesialis teknis yang menerjemahkan persyaratan bisnis pengguna sistem dan pembatas solusi teknis. Dia mendesain *database, input, output, screen* (layar/tampilan), jaringan, dan perangkat lunak komputer yang akan memenuhi persyaratan pengguna sistem.  
Contoh desainer sistem adalah :
  - a. Administrator database
  - b. Arsitek jaringan
  - c. Arsitek web
  - d. Artis grafik untuk antarmuka
  - e. Ahli keamanan
  - f. Spesialis teknologi, misal barcode
4. Pembangun sistem  
Spesialis teknis yang membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain yang dihasilkan oleh desainer.  
Contoh pembangun sistem adalah :
  - a. Programmer aplikasi
  - b. Programmer sistem
  - c. Programmer database
  - d. Administrator jaringan
  - e. Administrator keamanan
  - f. Webmaster
  - g. Software integrator, yang mengintegrasikan perangkat lunak dengan perangkat keras, jaringan dan perangkat-perangkat lunak lainnya.

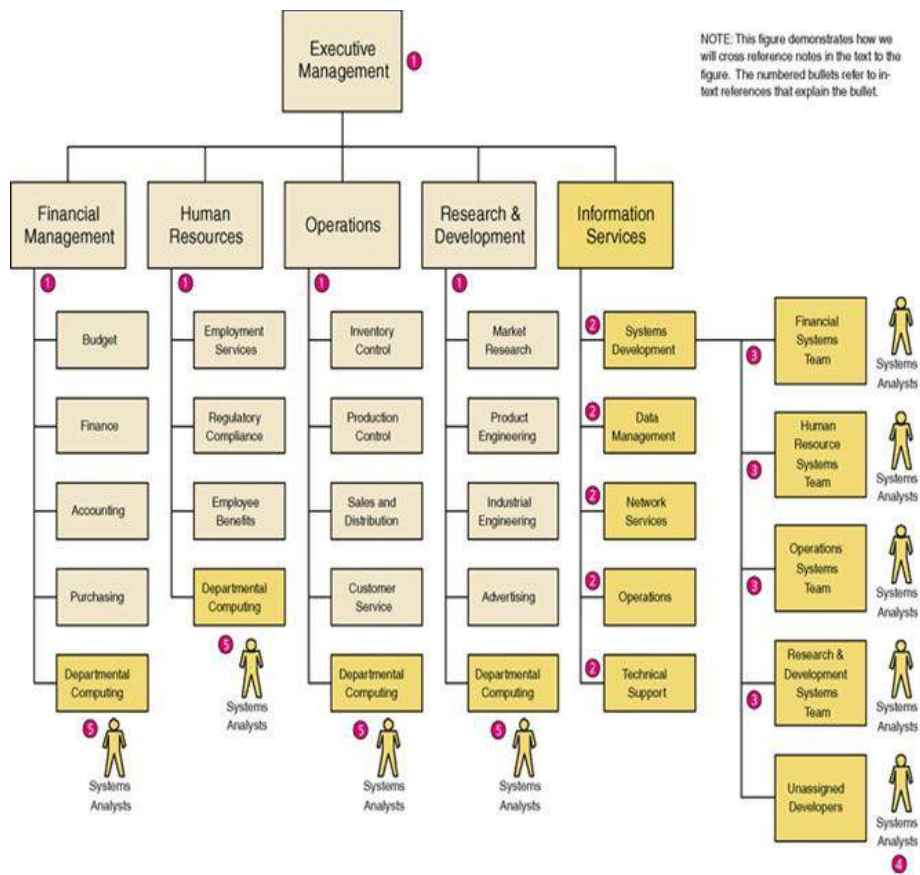
5. Analis sistem

Spesialis yang mempelajari masalah dan kebutuhan sebuah organisasi, untuk menentukan bagaimana orang, data, proses, dan teknologi informasi dapat mencapai kemajuan terbaik untuk bisnis.

Peran analis sistem adalah :

- a. Pemecah masalah
- b. Kesempatan untuk memperbaiki situasi meskipun tidak ada keluhan
- c. Petunjuk untuk mengubah situasi meskipun tidak ada orang yg mengeluh situasi sekarang

Peran sistem analis dalam suatu organisasi dapat digambarkan sebagai berikut (Whitten, dkk (2004)) : (gambar 8)



**Gambar 8.** Peran sistem analis dalam pengembangan dan pembangunan sistem informasi [1]

Keterangan gambar 2.2 :

Nomor 1 adalah Pemilik sistem

Nomor 2 adalah Desainer dan pembangun sistem

Nomor 3 dan 4 adalah analisis sistem, desainer dan pembangun sistem

Nomor 5 adalah Analisis sistem untuk departemen komputer yg lebih kecil

**2.2 Tugas**

Tentukan stakeholder yang terlibat dalam pengembangan sistem yang kelompok anda bangun !

## **BAB III**

### **PENGEMBANGAN SISTEM INFORMASI**

---

#### **Pertemuan ke-3**

##### **Tujuan :**

- 1. Mahasiswa memahami alasan pengembangan sistem informasi**
  - 2. Mahasiswa memahami tahapan-tahapan umum dalam pengembangan sistem informasi**
- 

#### **3.1 Definisi Pengembangan Sistem**

Pengembangan sistem adalah suatu proses penyusunan dan pembuatan suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sedang berjalan agar lebih optimal dalam membantu kinerja organisasi atau perusahaan.

Pengembangan sistem dilakukan karena alasan-alasan sebagai berikut :

1. Adanya permasalahan-permasalahan (*problems*) yang timbul di sistem yang lama. Permasalahan yang timbul dapat berupa :
  - a. Ketidakberesan sistem yang lama  
Ketidakberesan dalam sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan.
  - b. Pertumbuhan organisasi.  
Kebutuhan informasi yang semakin luas, volume pengolahan data semakin meningkat, perubahan prinsip akuntansi yang baru menyebabkan harus disusunnya sistem yang baru, karena sistem yang lama tidak efektif lagi dan tidak dapat memenuhi lagi semua kebutuhan informasi yang dibutuhkan manajemen.
2. Untuk meraih kesempatan-kesempatan  
Dalam keadaan persaingan pasar yang ketat, kecepatan informasi atau efisiensi waktu sangat menentukan berhasil atau tidaknya strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan-kesempatan dan peluang-peluang pasar, sehingga teknologi informasi perlu digunakan untuk meningkatkan penyediaan informasi agar dapat mendukung proses pengambilan keputusan yang dilakukan oleh manajemen.
3. Adanya instruksi dari pimpinan atau adanya peraturan pemerintah  
Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instruksi dari atas pimpinan ataupun dari luar organisasi, seperti misalnya peraturan pemerintah.

#### **3.2 Indikator Perlunya Pengembangan Sistem**

Adapun indikator-indikator di organisasi yang bisa menjadi alasan perlunya pengembangan sistem adalah :

1. Adanya keluhan pelanggan
2. Pengiriman barang yang sering tertunda
3. Pembayaran gaji yang terlambat
4. Laporan yang tidak tepat waktu
5. Isi laporan yang sering salah
6. Tanggung jawab yang tidak jelas
7. Waktu kerja yang berlebihan
8. Ketidakberesan kas
9. Produktivitas tenaga kerja yang rendah
10. Banyaknya pekerja yang menganggur
11. Kegiatan yang tumpang tindih
12. Tanggapan yang lambat terhadap pelanggan

13. Kehilangan kesempatan kompetisi pasar
14. Persediaan barang yang terlalu tinggi
15. Pemesanan kembali barang yang tidak efisien
16. Biaya operasi yang tinggi
17. File-file yang kurang teratur
18. Keluhan dari supplier karena tertundanya pembayaran
19. Tertundanya pengiriman karena kurang persediaan
20. Investasi yang tidak efisien
21. Peramalan penjualan dan produksi tidak tepat
22. Kapasitas produksi yang menganggur
23. Pekerjaan manajer yang terlalu teknis
24. DLL.

Jumlah personil yang dibutuhkan dalam pengembangan sistem biasanya disesuaikan dengan besar kecilnya sistem yang akan dibuat atau dikembangkan. Namun umumnya terdiri dari :

1. Manajer Analis Sistem
2. Ketua Analis Sistem
3. Analis Sistem Senior
4. Analis Sistem Junior
5. Pemrogram Aplikasi Senior
6. Pemrogram Aplikasi Junior

### **3.3 Prinsip-prinsip Pengembangan Sistem**

Dalam pengembangan suatu sistem informasi, harus mengikuti prinsip-prinsip pengembangan sistem informasi agar tujuan bisa tercapai. Adapun prinsip-prinsip dalam pengembangan sistem informasi adalah :

1. Sistem yang dikembangkan adalah untuk manajemen
2. Sistem yang dikembangkan adalah investasi modal yang besar.  
Maka setiap investasi modal harus mempertimbangkan 2 hal berikut ini :
  - a. Semua alternatif yang ada harus diinvestigasikan
  - b. Investasi yang terbaik harus bernilai
3. Sistem yang dikembangkan memerlukan orang yang terdidik
4. Tahapan kerja dan tugas-tugas yang baru dilakukan dalam proses pengembangan sistem
5. Proses pengembangan sistem tidak harus urut
6. Jangan takut membatalkan proyek
7. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem

### **3.4 Tujuan Pengembangan Sistem**

Tujuan utama pengembangan sistem adalah untuk memperbaiki sistem yang lama. Sistem baru diharapkan terjadi peningkatan dalam hal :

1. Kinerja, yang dapat diukur dari throughput dan respon time.  
*Throughput* : jumlah pekerjaan yang dapat dilakukan pada suatu saat tertentu  
*Respon time* : Rata-rata waktu tertunda di antara dua transaksi.
2. Kualitas informasi yang disajikan
3. Keuntungan (penurunan biaya). Berhubungan dengan jumlah sumber daya yang digunakan
4. Kontrol (pengendalian)
5. Efisiensi
6. Pelayanan

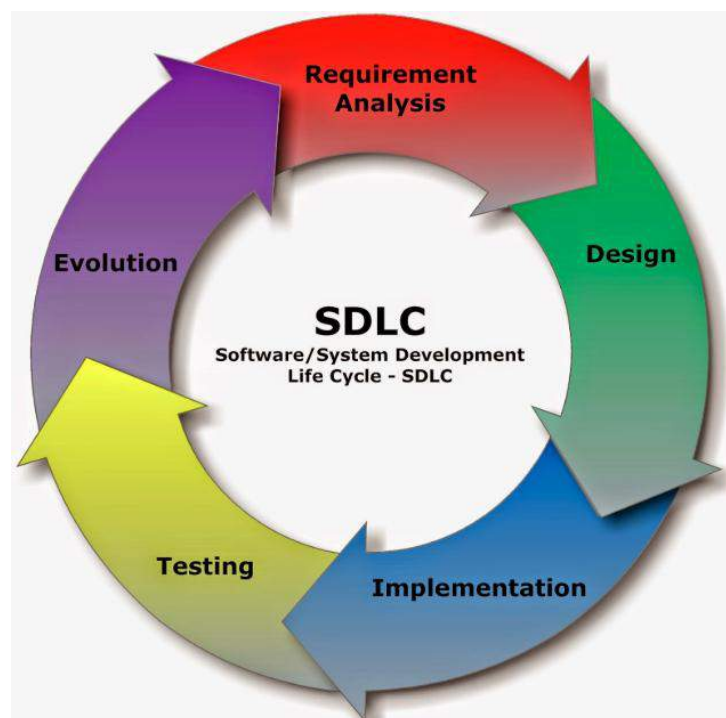


Bila dalam operasi sistem yang sudah dikembangkan masih timbul permasalahan-permasalahan yang tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke proses yang pertama. Siklus ini disebut dengan Siklus Hidup suatu Sistem.

### 3.5 Siklus Hidup Pengembangan Sistem

Siklus Hidup Pengembangan Sistem dapat didefinisikan sebagai serangkaian aktivitas yang dilaksanakan oleh profesional dan pemakai sistem informasi untuk mengembangkan dan mengimplementasikan sistem informasi.

Ada banyak model dalam pengembangan sistem informasi, seperti *waterfal*, *prototyping*, *rapid application development (RAD)*, *incremental* dan lain-lain. Namun, secara garis besar aktivitas yang dilakukan dalam setiap tahapan terbagi menjadi enam seperti gambar 9 di bawah ini :



**Gambar 9.** Siklus hidup pengembangan sistem informasi [1]

Siklus hidup pengembangan sistem informasi terbagi atas enam fase, yaitu :

1. Analisis sistem/*Requirement Analysis*  
Dalam fase ini dilakukan :
  - a. Dibentuk suatu struktur kerja yang akan memberikan gambaran mengenai sistem yang sedang berjalan dan harapan pemakai sistem pada sistem yang akan dibangun. Gambaran umum ini bisa berupa deskripsi mengenai sistem yang sedang berjalan. Bisa berupa uraian atau juga bisa berupa flowchart yang menggambarkan sistem yang sedang berjalan saat ini.
  - b. Suatu sistem yang diusulkan harus layak, yaitu sistem ini harus memenuhi kriteria-kriteria sebagai berikut :
    - 1) Kelayakan teknis untuk melihat apakah sistem yang diusulkan dapat dikembangkan dan diimplementasikan dengan menggunakan teknologi yang ada atau apakah teknologi yang baru dibutuhkan.

- 2) Kelayakan ekonomis untuk melihat apakah dana yang tersedia cukup untuk mendukung estimasi biaya untuk sistem yang diusulkan.
  - 3) Kelayakan legal untuk melihat apakah ada konflik antara sistem yang sedang dipertimbangkan dengan kemampuan perusahaan untuk melaksanakan kewajibannya secara legal.
  - 4) Kelayakan operasional untuk melihat apakah prosedur dan keahlian pegawai yang ada cukup untuk mengoperasikan sistem yang diusulkan atau apakah diperlukan penambahan/pengurangan prosedur dan keahlian. Definisi dan identifikasi masalah, tujuan pengembangan sistem, kebutuhan yang diperlukan selama proses pengembangan sistem, prioritas sistem yang akan dikembangkan dan kendala-kendala yang akan dihadapi dalam pengembangan sistem;
- c. Identifikasi masalah dan solusi kebutuhan yang diperlukan selama proses pengembangan sistem.
- d. Dokumentasi hasil analisa yang berisi penemuan-penemuan dan rekomendasi serta alasan untuk mengembangkan suatu sistem baru. Dituangkan dalam analisa kebutuhan fungsional dan nonfungsional.
- 1) Kebutuhan Fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan Fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.
  - 2) Kebutuhan non fungsional adalah tipe kebutuhan yang berisi properti pe laku yang dimiliki oleh sistem. Kebutuhan non fungsional dapat dicari melalui beberapa komponen :
    - Operasional : bagaimana system baru beroperasi dilihat dari aspek hardware dan software
    - Kinerja : seberapa bagus kinerja software baru dalam mengolah data, menampilkan informasi, dan seberapa efisien dalam menangani keseluruhan proses bisnis
    - Keamanan : password dan hardware untuk pengamanan system.
2. Perancangan / *Design System*  
Yang dilakukan dalam fase ini :
- a. Pada fase ini analis sistem mulai merancang proses dengan mengidentifikasi laporan-laporan dan output yang akan dihasilkan oleh sistem yang diusulkan. Data masing-masing laporan ditentukan. Biasanya, perancang sistem membuat perancangan sistem tentang output, input, proses, kendali, *database* dan teknologi akan dirancang.
  - b. Menerangkan data yang akan dimasukkan, dihitung atau disimpan. Perancang sistem memilih struktur file dan alat penyimpanan seperti *database*, *cloud computing* atau bahkan filefile dokumen. Prosedur-prosedur yang ditulis menjelaskan bagaimana data diproses untuk menghasilkan output.
3. Implementasi sistem  
Yang dilakukan pada fase ini adalah pembangunan aplikasi menggunakan *software* yang diusulkan.
4. Pengujian / *Testing*
5. Evolusi



---

## BAB IV ANALISIS SISTEM

---

### Pertemuan ke-4,5,6,7

#### Tujuan :

1. Mahasiswa memahami tahapan-tahapan dalam analisis sistem
  2. Mahasiswa mampu melakukan analisis sistem pada kasus yang diangkat
- 

#### 4.1 Aktifitas Pada Fase Analisis Sistem

Analisis sistem merupakan tahapan pertama yang dilakukan dalam proses pengembangan perangkat lunak.

Aktifitas yang dilakukan dalam fase analisis sistem :

1. Dibentuk suatu struktur kerja yang akan memberikan gambaran mengenai sistem yang sedang berjalan dan harapan pemakai sistem pada sistem yang akan dibangun. Gambaran umum ini bisa berupa deskripsi mengenai sistem yang sedang berjalan. Bisa berupa uraian atau juga bisa berupa flowchart yang menggambarkan sistem yang sedang berjalan saat ini.

#### Contoh kasus pengembangan sistem informasi perpustakaan AMIKOM Yogyakarta.

Deskripsi kasus : Perusahaan software "X" akan mengembangkan sistem informasi perpustakaan pada Universitas AMIKOM Yogyakarta. Adapun visi misi yang dimiliki oleh universitas tersebut adalah :

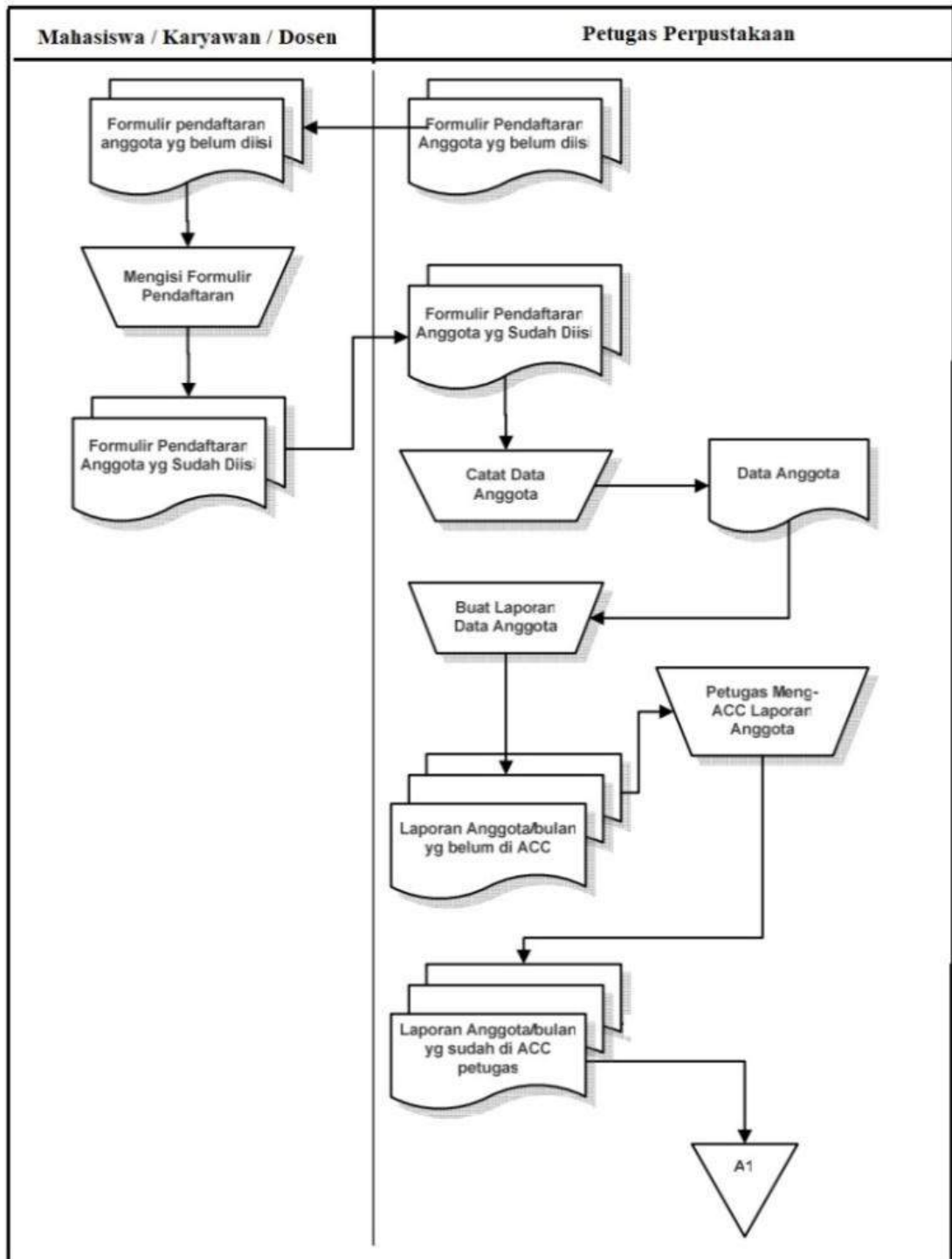
- Visi, Menjadi pusat penyedia bahan pustaka ilmu-ilmu komputer dan sistem informasi yang lengkap, efektif, akurat dan handal bagi mahasiswa, karyawan dan anggota perpustakaan AMIKOM.
- Misi , Meningkatkan minat akses masyarakat pada perpustakaan ini. Menyediakan sumber bahan referensi ilmu-ilmu komputer dan sistem informasi yang lengkap.

Tujuan pembangunan sistem

1. Pertumbuhan anggota pertahun sebesar 10 % pertahun.
2. Peningkatan akses anggota sebesar 2% perbulan
3. Peningkatan sumber daya buku dan referensi sebesar 3 % perbulan

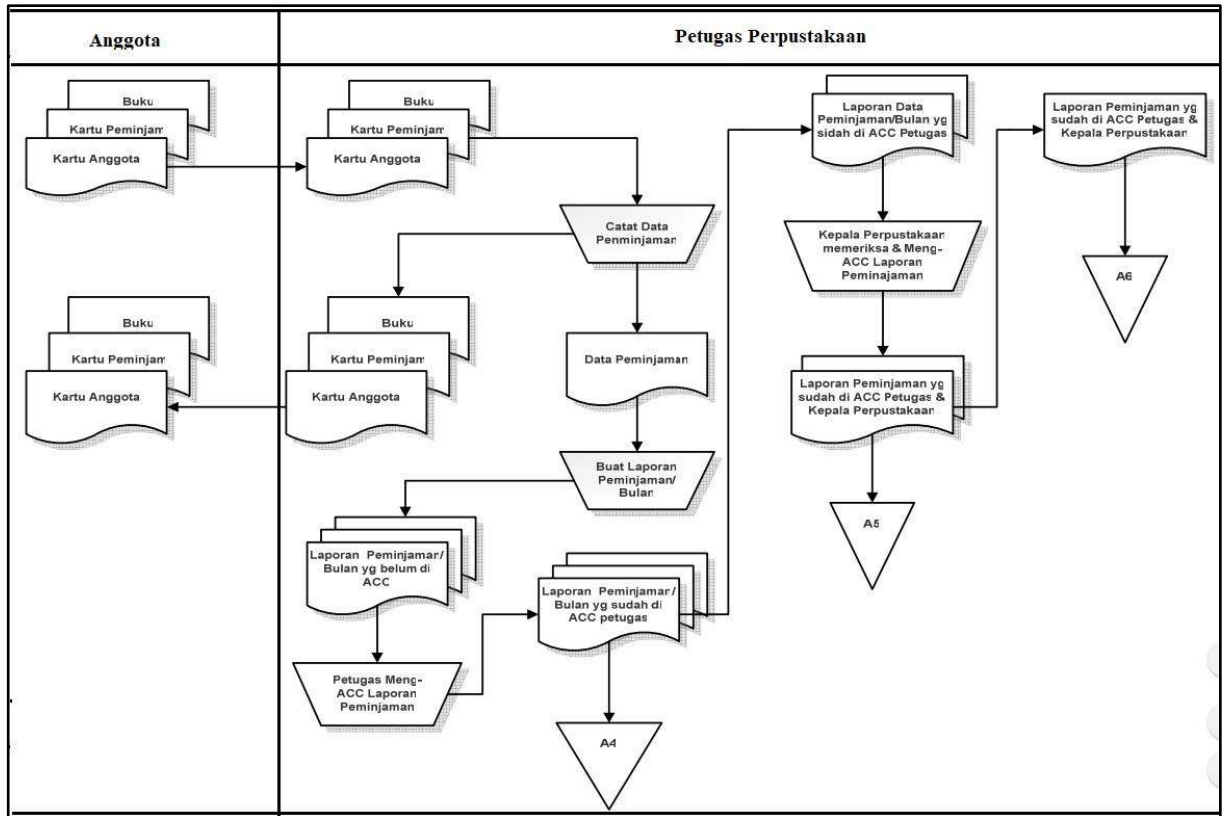
Dalam menjalankan kegiatannya, Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.

Penggambaran flowchart sistem yang berjalan pada pendaftaran anggota adalah :



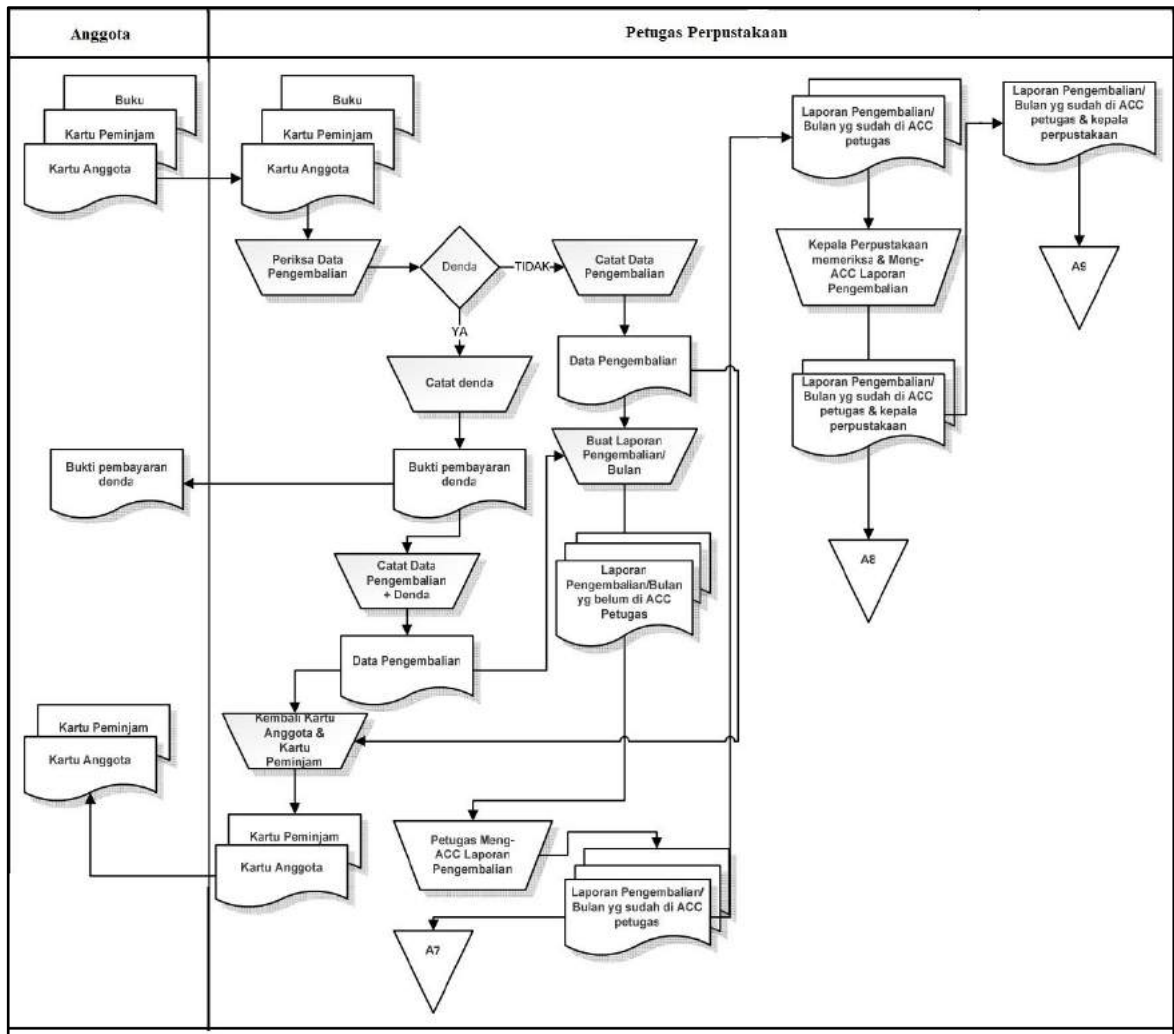
Gambar 10. Flowchart sistem yang berjalan pada pendaftaran anggota

Penggambaran flowchart sistem yang berjalan pada peminjaman buku adalah :



**Gambar 11.** Flowchart sistem yang berjalan pada peminjaman buku

Penggambaran flowchart sistem yang berjalan pada pengembalian buku adalah :



**Gambar 12.** Flowchart sistem yang berjalan pada pengembalian buku

2. Sistem yang akan dikembangkan dievaluasi dan dipisahkan berdasarkan prioritasnya. Sistem dengan prioritas tertinggi (sangat penting) akan dipilih untuk pengembangan. Suatu sistem yang diusulkan harus layak, yaitu sistem ini harus memenuhi kriteria-kriteria sebagai berikut :
  - a. Kelayakan teknis untuk melihat apakah sistem yang diusulkan dapat dikembangkan dan diimplementasikan dengan menggunakan teknologi yang ada atau apakah teknologi yang baru dibutuhkan.  
Beberapa pertanyaan yang bisa digunakan untuk mengukur kelayakan teknik pada sistem yang akan dikembangkan :
    - 1) Apakah teknologi yang dibutuhkan sudah tersedia?
    - 2) Apakah teknologi yang akan digunakan ini dapat berintegrasi dengan teknologi yang sudah ada?
    - 3) Apakah sistem yang sudah ada dapat dikonversikan ke sistem dengan teknologi baru?
    - 4) Apakah organisasi memiliki orang yang menguasai teknologi baru ini?

Analisa kelayakan teknologi dari kasus pengembangan sistem informasi perpustakaan AMIKOM adalah :

**1) Apakah teknologi yang dibutuhkan sudah tersedia?**

Berdasarkan analisis teknologi dari segi hardware, Perpustakaan sudah memiliki perangkat/komponen yang layak dioperasikan dan jaringan internet yang siap untuk digunakan.

**2) Apakah teknologi yang akan digunakan ini dapat berintegrasi dengan teknologi yang sudah ada?**

Teknologi yang lama masih menggunakan cara tradisional sehingga ketika terjadi konversi tidak akan memakan biaya yang banyak, hanya butuh tenaga untuk menginputkan semua data yang akan disimpan.

**3) Apakah sistem yang sudah ada dapat dikonversikan ke sistem dengan teknologi baru?**

Sistem yang lama tidak dapat dikonversikan ke sistem yang baru dikarenakan perbedaan mekanisme pengolahan datanya. Data yang ada di sistem yang lama harus diinputkan ke dalam database / tempat penyimpanan aplikasi di sistem yang baru.

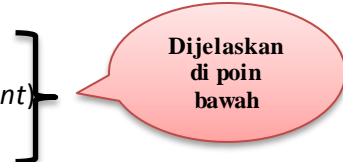
**4) Apakah organisasi memiliki orang yang menguasai teknologi baru ini?**

Belum ada tenaga yang mengerti mengenai fisik terutama coding dari sistem yang dibuat. Namun beberapa staf mampu mengoperasikan komputer dan familiar dengan computer sehingga bisa dijadikan staf yang bertugas untuk menginputkan data, mengolah transaksi, mencetak laporan dan melakukan backup data. Jika sistem bermasalah pada segi coding maka permasalahan kan dilaporkan ke pembuat program

- b. Kelayakan ekonomis untuk melihat apakah dana yang tersedia cukup untuk mendukung estimasi biaya untuk sistem yang diusulkan.

Menurut Syaifullah dan Jony Widiyanto (2014) kelayakan ekonomi bisa diukur menggunakan analisis biaya dan manfaat (*cost benenefit analysis*). Tujuan dari analisis ini adalah untuk memberikan gambaran kepada pengguna apakah manfaat yang diperoleh dari sistem baru "lebih besar" dibandingkan dengan biaya yang dikeluarkan. Pada analisis biaya dan manfaat, ada beberapa metode kuantitatif yang digunakan untuk menemukan standar kelayakan sistem yang akan dibangun seperti :

- 1) Metode Periode Pengembalian (*Payback Period*)
- 2) Metode Pengembalian Investasi (*Return on Investment*)
- 3) Metode Nilai Sekarang Bersih (*Net Present Value*)



Ketiga metode tersebut memiliki kesamaan, yaitu sama-sama menganalisa komponen biaya yang dikeluarkan saat sistem dibangun dan manfaat yang bisa didapatkan.

Biaya yang berhubungan dengan pengembangan sistem informasi dapat diklasifikasikan ke dalam 4 katagori utama, yaitu :

a) Biaya pengadaan (*procurement cost*)

Adalah semua biaya yang terjadi sehubungan dengan memperoleh perangkat keras. Yang termasuk biaya pengadaan diantaranya adalah :

- 1) Biaya konsultasi pengadaan perangkat keras.
- 2) Biaya pembelian atau sewa beli (*leasing*) perangkat keras.
- 3) Biaya instalasi perangkat keras.
- 4) Biaya ruangan untuk perangkat keras (perbaikan ruangan, pemasangan AC).
- 5) Biaya modal untuk pengadaan perangkat keras.
- 6) Biaya yang berhubungan dengan manajemen dan satff untuk pengadaan perangkat keras.

b) Biaya persiapan operasi (*start-up cost*)

Adalah semua biaya untuk membuat sistem siap untuk dioperasikan. Yang termasuk biaya persiapan diantaranya adalah :

- 1) Biaya pembelian perangkat lunak sistem.
  - 2) Biaya instalasi peralatan komunikasi (misal sambungan telpon).
  - 3) Biaya persiapan personil.
  - 4) Biaya reorganisasi.
  - 5) Biaya manajemen staff yang dibutuhkan dalam kegiatan persiapan operasi.
- c) Biaya proyek (*project-related cost*)

Adalah semua biaya untuk mengembangkan sistem termasuk penerapannya. Yang termasuk biaya proyek diantaranya :

- 1) Biaya dalam tahap analisis sistem mencakup :
  - Biaya untuk pengumpulan data
  - Biaya dokumentasi (kertas, foto copy, dll)
  - Biaya Rapat
  - Biaya staff analis
  - Biaya manajemen yang berhubungan dengan tahap analisis sistem
- 2) Biaya dalam tahap design sistem mencakup :
  - Biaya dokumentasi
  - Biaya Rapat
  - Biaya staff analis
  - Biaya programmer
  - Biaya pembelian perangkat lunak aplikasi
  - Biaya manajemen yang berhubungan dengan tahap design sistem
- 3) Biaya dalam tahap penerapan sistem mencakup :
  - Biaya pembuatan formulir baru
  - Biaya konversi data
  - Biaya latihan personil
  - Biaya manajemen yang berhubungan dengan tahap penerapan sistem

Catatan : Bila menggunakan konsultan ada biaya tambahan honor konsultan.

- d) Biaya operasi (*ongoing cost*) dan biaya perawatan (*maintenance cost*)
- Biaya operasi adalah biaya-biaya yang dikeluarkan untuk mengoperasikan sistem supaya sistem dapat beroperasi. Biaya perawatan adalah biaya yang dikeluarkan untuk merawat sistem dalam masa operasinya. Yang termasuk biaya operasi dan perawatan diantaranya :

- 1) Biaya personil (operator, bagian administrasi, pustakawan data, pengawas data)
- 2) Biaya overhead (pemakaian telpon, listrik, asuransi, keamanan, suplies)
- 3) Biaya perawatan perangkat keras
- 4) Biaya perawatan perangkat lunak (modifikasi program, penambahan modul program)
- 5) Biaya perawatan peralatan dan fasilitas
- 6) Biaya manajemen yang terlibat dalam operasi sistem
- 7) Biaya kontrak untuk konsultan selama operasi sistem
- 8) Biaya depresiasi (penyusutan)

Catatan : Biaya ini rutin keluar selama sistem digunakan (seumur hidup sistem)

**Metode Periode Pengembalian (*Payback Period*)**

Metode ini adalah uji kuantitatif yang digunakan untuk menghitung jangka waktu yang diperlukan untuk membayar kembali biaya investasi dalam pembuatan aplikasi yang telah dikeluarkan. Penilaian kelayakan untuk *payback* adalah :

$$periode = \frac{investasi}{proceed} \times 1 \text{ tahun}$$

- a. Layak jika waktu pengembalian lebih kecil dari umur investasi.
- b. Tidak layak jika waktu pengembalian lebih besar dari umur investasi.

Contoh perhitungan PP pada kasus pengembangan sistem informasi perpustakaan AMIKOM : **(kasus diambil dari jurnal yang ditulis oleh Pratama, dkk)**

Penilaian proyek investasi menggunakan metode ini didasarkan pada lamanya investasi tersebut dapat tertutup dengan aliran-aliran kas masuk dan faktor bunga tidak dimasukkan dalam perhitungan ini. Berikut adalah penjabaran Payback Period dari Software.

Diketahui jumlah investasi awal pada tahun ke-0 adalah sebesar Rp 524.250.000. Pada tabel dibawah ini dijelaskan bahwa pada tahun ke-1 jumlah proceed kumulatifnya sudah bisa menutup investasi awal. Jumlah proceed kumulatif di tahun ke-1 adalah sebesar Rp 237.000.000 dan di tahun ke2 sebesar Rp 497.700.000

### Rincian Biaya dan Manfaat Software

Rincian Biaya R-Software	Tahun 0	Tahun 1	Tahun 2	Tahun 3
<b>Biaya Pengadaan Hardware</b>				
Biaya Pengadaan Server	Rp 500.000.000	Rp -	Rp -	Rp -
<b>Biaya Persiapan Operasional</b>				
Biaya Rekrutment SDM	Rp 144.000.000	Rp -	Rp -	Rp -
<b>Biaya Project Related</b>				
Tidak ada	Rp -	Rp -	Rp -	Rp -
<b>Biaya Operasional dan Perawatan</b>				
Biaya Lisensi Antivirus	Rp -	Rp 50.000.000	Rp 55.000.000	Rp 60.500.000
Stasionery	Rp -	Rp 6.500.000	Rp 7.150.000	Rp 7.865.000
Biaya Inventaris Kantor	Rp -	Rp 30.000.000	Rp 33.000.000	Rp 36.300.000
Biaya Pemeliharaan Perangkat	Rp -	Rp 44.000.000	Rp 48.400.000	Rp 53.240.000
Biaya Maintenance Data Centre	Rp -	Rp 20.000.000	Rp 22.000.000	Rp 24.200.000
Biaya Gaji Karyawan		Rp 144.000.000	Rp 158.400.000	Rp 174.240.000
Biaya Internet	Rp -	Rp 3.600.000.000	Rp 3.960.000.000	Rp 4.356.000.000
<b>TOTAL BIAYA</b>	<b>Rp 644.000.000</b>	<b>Rp 3.844.500.000</b>	<b>Rp 4.228.950.000</b>	<b>Rp 4.651.845.000</b>
<b>Rincian Manfaat R-Software</b>				
<b>Biaya Manfaat</b>				
Pendapatan per Tahun	Rp -	Rp 4.400.000.000	Rp 4.400.000.000	Rp 4.840.000.000
<b>TOTAL MANFAAT</b>	<b>Rp -</b>	<b>Rp 4.400.000.000</b>	<b>Rp 4.400.000.000</b>	<b>Rp 4.840.000.000</b>
<b>PROCEED</b>	<b>Rp (644.000.000)</b>	<b>Rp 555.500.000</b>	<b>Rp 171.050.000</b>	<b>Rp 188.155.000</b>

Payback Period untuk proyek ini adalah

$$PP = 1 + \frac{524.250.000 - 237.000.000}{497.700.000 - 237.000.000} \times 1 \text{ tahun}$$

= 2,1 atau kurang lebih 2 tahun 1 bulan

Hal ini dapat dikatakan layak karena waktu pengembalian lebih kecil dari umur investasi maksimal yaitu 3 tahun.

#### Metode Pengembalian Investasi (Return on Investment)

Metode pengembalian investasi digunakan untuk mengukur presentase manfaat yang dihasilkan proyek dibanding dengan biaya yang dikeluarkan. *Return on Investmen* (ROI) dari suatu proyek dapat dihitung dengan rumus :



$$ROI = \frac{(total\ manfaat - total\ biaya)}{total\ biaya} \times 100\%$$

Penilaian kelayakan untuk ROI :

- Layak jika  $ROI > 0$
- Tidak layak jika  $ROI < 0$

Berikut adalah penjabaran ROI dari PHP dan My-SQL.

Diketahui bahwa total manfaat dari Software adalah Rp14.564.000.000 Sementara total biaya dari PHP adalah Rp13.779.530.000 ROI untuk proyek ini adalah

$$ROI = \frac{(14.564.000.000 - 13.779.530.000)}{13.779.530.000} \times 100\% = 6\%$$

Apabila suatu proyek investasi mempunyai ROI lebih besar dari 0 maka proyek tersebut dapat diterima. Pada proyek ini, nilai ROI nya adalah 5,69 atau 6%. Ini berarti proyek ini dapat diterima dengan memberikan keuntungan sebesar 6% dari total biaya investasinya.

#### Metode Nilai Sekarang Bersih (Net Present Value)

Metode nilai sekaang bersih merupakan metode yang memperhatikan nilai waktu dari uang. Suku bunga diskonto mempengaruhi *proceed* atau arus dari uangnya. *Net present value* (NPV) dapat dihitung dari selisih nilai proyek pada awal tahun dikurangi dengan *proceed* tiap tahun yang dinilai uangkan ketahun awal dengan tingkat bunga diskonto.

Kriteria NPV :

- NPV > 0 *Feasible*
- NPV = 0 *Indifferent*
- NPV < 0 *Unfeasible*

Rumus untuk menghitung NPV yaitu :

$$NPV = -nilai\ proyek + \frac{procced1}{(1+i)^1} + \frac{procced2}{(1+i)^2} + \frac{procced3}{(1+i)^3}$$

Keterangan :

- NPV = Net Present Value
- I=Tingkat bunga diskonto diperhitungkan
- n = umur proyek investasi
- Procced* = Selisih biaya dan manfaat

Metode NPV memperhatikan nilai waktu dari uang. Metode ini menggunakan tingkat suku bunga yang mempengaruhi arus dari uang. Dalam metode ini, bunga yang diketahui berasal dari bunga Bank Indonesia per 6 Februari 2015 yaitu sebesar 6,93% dengan masa umur proyek selama 3 tahun. Berikut adalah penjabaran NPV dari Criterion dan Software. Contoh perhitungan PP pada kasus pengembangan sistem informasi perpustakaan AMIKOM :

Diketahui bahwa total *proceed*/ arus kas masuk dari PHP adalah Rp 784.470.000. NPV untuk proyek ini adalah

$$NPV = -nilai\ proyek + \frac{procced1}{(1+i)^1} + \frac{procced2}{(1+i)^2} + \frac{procced3}{(1+i)^3}$$
$$NPV = -524.250.000 + \frac{237.000.000}{(1+6,9\%)^1} + \frac{260.700.000}{(1+6,9\%)^2} + \frac{286.770.000}{(1+6,9\%)^3}$$



$$NPV = 160.401.380$$

Dari hasil perhitungan diatas, diketahui bahwa nilai NPV untuk investasi PHP adalah sebesar Rp 160.401.380, ini berarti bahwa nilai NPV proyek tersebut positif atau  $NPV > 0$  sehingga proyek ini dapat diterima atau layak dilaksanakan.

- c. Kelayakan legal untuk melihat apakah ada konflik antara sistem yang sedang dipertimbangkan dengan kemampuan perusahaan untuk melaksanakan kewajibannya secara legal.

Software yang digunakan yaitu PHP dan My-SQL merupakan program open source sehingga tidak diperlukan pembelian lisensi dan menggunakan hak cipta publik. Dari segi kelayakan hukum, AMIKOM dapat menerapkan software tersebut dan tidak akan terkendala masalah hukum terutama hak cipta karena masing-masing software open source.

- d. Kelayakan operasional untuk melihat apakah prosedur dan keahlian pegawai yang ada cukup untuk mengoperasikan sistem yang diusulkan atau apakah diperlukan penambahan/pengurangan prosedur dan keahlian. Beberapa pertanyaan yang bisa digunakan untuk mengukur kelayakan operasional pada sistem yang akan dikembangkan :
- 1) Apakah sistem dapat memenuhi tujuan organisasi untuk mendapatkan informasi yang dibutuhkan?
  - 2) Apakah sistem dapat diorganisasikan untuk menghasilkan informasi pada saat yang tepat untuk setiap orang yang membutuhkannya?
  - 3) Apakah sistem baru memerlukan restrukturisasi organisasi dan bagaimana akibat strukturisasi ini terhadap orang-orang yang ada di organisasi?
  - 4) Apakah diperlukan pelatihan atau pelatihan ulang?
  - 5) Apakah personil di dalam organisasi dapat memenuhi kriteria untuk sistem baru?

Analisa kelayakan operasional dari kasus pengembangan sistem informasi perpustakaan AMIKOM adalah :

**1) Apakah sistem dapat memenuhi tujuan organisasi untuk mendapatkan informasi yang dibutuhkan?**

Sistem yang dibangun dapat merekap laporan-laporan yang akan digunakan oleh organisasi sebagai dasar informasi dan pertimbangan dalam pengambilan kebijakan. Laporan yang dihasilkan dari sistem yang akan dibangun antara lain :

- a. Sistem dapat menampilkan laporan anggota registrasi, aktif maupun pasif di setiap bulannya dalam bentuk diagram batang
- b. Sistem dapat menampilkan laporan perkembangan anggota berdasarkan statusnya anggota registrasi, anggota aktif dan anggota pasif
- c. Sistem dapat menampilkan laporan perkembangan anggota berdasarkan PRODI nya
- d. Sistem dapat menampilkan laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang
- e. Sistem dapat menampilkan laporan jumlah buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang
- f. Petugas bisa melihat laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang dengan login terlebih dahulu
- g. Petugas bisa melihat laporan buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang

**2) Apakah sistem dapat diorganisasikan untuk menghasilkan informasi pada saat yang tepat untuk setiap orang yang membutuhkannya?**

sistem yang dibangun dapat menghasilkan informasi pada saat yang tepat untuk setiap orang

yang membutuhkannya. Informasi yang bisa dihasilkan oleh sistem nantinya seperti :

- a. Petugas : Sistem yang dibangun bisa menampilkan laporan anggota registrasi, aktif maupun pasif di setiap bulannya ; laporan perkembangan anggota berdasarkan statusnya anggota registrasi, anggota aktif dan anggota pasif; menampilkan laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya ; laporan jumlah buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya ; laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya; laporan buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya.
- b. Anggota perpustakaan bisa mendapatkan informasi ketersediaan buku dan koleksi yang sedang dicari serta letak penempatan buku tersebut.

**3) Apakah sistem baru memerlukan restrukturisasi organisasi dan bagaimana akibat restrukturisasi ini terhadap orang-orang yang ada di organisasi?**

Tidak ada pergantian atau pengurangan jumlah petugas di perpustakaan, hanya ada pembagian tugas di beberapa bagian, mengingat beberapa tugas sudah bisa ditangani oleh sistem baru yang akan diimplementasikan. Misalnya, petugas yang mencatat transaksi peminjaman dan pengembalian awalnya berjumlah 4 orang bisa dikurangi. Sistem baru yang diusulkan hanya memerlukan 2 orang petugas untuk melayani transaksi peminjaman dan pengembalian buku. Dengan 2 orang petugas saja maka waktu menganggur bisa diperkecil. Sisa petugas lainnya bisa difungsikan untuk pekerjaan lain.

**4) Apakah diperlukan pelatihan atau pelatihan ulang?**

Diperlukan pelatihan cara menggunakan software kepada para petugas.

**5) Apakah personil di dalam organisasi dapat memenuhi kriteria untuk sistem baru?**

Petugas di Perpustakaan AMIKOM sudah familiar dengan penggunaan komputer, sehingga memenuhi kriteria untuk pengimplementasian sistem yang baru

3. Identifikasi masalah dan solusi kebutuhan yang diperlukan selama proses pengembangan sistem.
4. Dokumentasi hasil analisa yang berisi penemuan-penemuan dan rekomendasi serta alasan untuk mengembangkan suatu sistem baru. Dituangkan dalam analisa kebutuhan fungsional dan nonfungsional.
  - a. Kebutuhan Fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan oleh sistem. Kebutuhan Fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan oleh sistem.
  - b. Kebutuhan non fungsional adalah tipe kebutuhan yang berisi properti pelaku yang dimiliki oleh sistem. Kebutuhan non fungsional dapat dicari melalui beberapa komponen :
    - Operasional : bagaimana system baru beroperasi dilihat dari aspek hardware dan software
    - Kinerja : seberapa bagus kinerja software baru dalam mengolah data, menampilkan informasi, dan seberapa efisien dalam menangani keseluruhan proses bisnis
    - Keamanan : password dan hardware untuk pengamanan system.

Contoh Identifikasi masalah, penentuan solusi kebutuhan dan dokumentasi analisa kebutuhan fungsional dan nonfungsional pada kasus pengembangan sistem informasi perpustakaan AMIKOM.

**Tabel 1.** Analisa identifikasi masalah, solusi dan analisa kebutuhan fungsional dan nonfungsional

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencari koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.

Identifikasi masalah dan penentuan solusi dari permasalahan di atas adalah :

No.	Identifikasi Masalah	Solusi Yang Diusulkan	Kebutuhan Fungsional	Kebutuhan Non fungsional
1.	Dalam 1 hari jumlah rata-rata transaksi peminjaman dan pengembalian sebanyak 150 transaksi. 1 transaksi dapat dilayani dalam waktu kurang lebih 4 menit. Dengan jumlah petugas sebanyak 4 orang, maka dalam 1 harinya rata-rata 1 orang petugas akan menghabiskan waktu sebanyak $(150/4)*4$ menit = 152 menit dalam melayani transaksi peminjaman dan pengembalian. Jumlah jam kerja petugas sebesar 450 menit dibandingkan dengan rata-rata tiap petugas melayani transaksi dalam sehari yaitu 152 menit, dapat disimpulkan terjadi waktu menganggur yang cukup banyak yaitu $450 - 152 = 298$ menit dalam seharinya.	Dibuat suatu system yang mampu mengefektifkan kinerja petugas perpustakaan. Dengan asumsi 150 transaksi perhari dan jam kerja 450 menit perhari maka system baru yang diusulkan hanya memerlukan 2 orang petugas untuk melayani transaksi peminjaman dan pengembalian buku. Dengan 2 orang petugas saja maka waktu menganggur bisa diperkecil yaitu $450 - ((150/2)* 4) = 150$ menit.	Salah satu pengguna system adalah level petugas. Petugas memiliki hak akses untuk melakukan pengolahan data transaksi peminjaman dan pengembalian buku	Sistem mampu memproses setiap transaksi dalam waktu 2 menit
2.	Dari hasil perhitungan response time didapatkan angka sebesar 7,84 menit atau kurang lebih 8 menit. Ini berarti dalam melakukan pelayanan,	a) Sistem yang diusulkan mampu menangani transaksi peminjaman atau pengembalian buku di setiap transaksinya dalam waktu 2 menit.	-	a) Penambahan jumlah bandwidth hingga 500 Mbps b) Menggunakan software

	<p>petugas memerlukan waktu sekitar 8 menit dalam setiap transaksi peminjaman atau pengembalian buku. Response time yang cukup lama berakibat pada beberapa hal yaitu :</p> <p>a) sering terjadinya antrian pada jam-jam tertentu misalnya di jam istirahat. Hal ini membuat anggota seringkali mengeluh dan memilih membatalkan proses peminjaman maupun pengembalian buku.</p> <p>b) Menurunnya jumlah anggota dan jumlah transaksi peminjaman maupun pengembalian buku</p>	<p>b) Melakukan penambahan bandwidth.</p> <p>c) Menggunakan database My. SQL dalam penyimpanan datanya.</p> <p>d) Sistem dapat melakukan backup otomatis dalam setiap bulan.</p>		<p>MySQL untuk melakukan manipulasi data di database</p> <p>c) Sistem dapat melakukan backup data 1 bulan sekali</p> <p>d) Minimum hardware yang dipakai ....</p> <p>e) Sistem yang dibangun menggunakan software PHP</p>
3.	<p>Belum adanya fasilitas pencarian buku yang bisa digunakan oleh anggota perpustakaan</p>	<p>a) Sistem baru yang diusulkan terdapat fitur untuk melakukan pencarian buku berdasarkan judul, pengarang, penerbit.</p> <p>b) Hasil pencarian berisi informasi yang berisi kode buku, judul, pengarang, penerbit, tahun terbit, jumlah buku yang tersedia dan bisa dipinjam, kode rak penyimpanan.</p>	<p>a) Anggota adalah salah satu pengguna aplikasi yang bisa menggunakan pencarian data buku tanpa harus melakukan login terlebih dahulu</p> <p>b) Sistem memiliki suatu fitur yang dapat melakukan pencarian data buku berdasarkan judul, pengarang, penerbit.</p> <p>c) Sistem dapat menampilkan hasil pencarian buku oleh anggota yang berisi informasi kode buku, judul, pengarang, penerbit, tahun terbit, jumlah buku yang tersedia dan bisa dipinjam, kode rak penyimpanan.</p>	-
4.	<p>Sistem belum ada fasilitas penggolongan data buku untuk tiap</p>	<p>a) Sistem baru sudah terdapat fitur untuk melakukan penggolongan buku</p>	<p>a) Sistem dapat menggolongkan jenis buku berdasarkan PRODI, tahun</p>	-

	<p>raknya. Hal ini menyulitkan petugas dalam penempatan buku baru.</p>	<p>berdasarkan PRODI. b) Masing-masing jenis buku sudah ditempatkan di rak khusus dengan aturan penomoran : ISBN/PRODI/TAHUN_TERBIT/BUKU_KE</p>	<p>terbit, dan kategori buku. b) Sistem dapat melakukan pengolahan data rak (menambah, mengedit dan menghapus data rak) serta melakukan pengolahan data jenis buku yang disimpan di masing-masing rak. c) Sistem dapat melakukan pengesetan secara otomatis untuk setiap kode buku baru yang diinputkan. Kode buku diatur dengan aturan penomoran : ISBN/PRODI/TAHUN_TERBIT/BUKU_KE</p>	
5.	<p>sistem belum bisa menampilkan data buku beserta pengarang dan penerbitnya. Akibatnya petugas sering kesulitan dalam melakukan reorder buku.</p>	<p>a) Sistem baru yang diusulkan terdapat fitur untuk melakukan pencarian buku berdasarkan judul, pengarang, penerbit. b) Hasil pencarian berisi informasi yang berisi kode buku, judul, pengarang, penerbit, tahun terbit, jumlah buku yang tersedia dan bisa dipinjam, kode rak penyimpanan.</p>	<p>a) Untuk menggunakan aplikasi, petugas harus login terlebih dahulu b) Petugas bisa melakukan olah data (menginputkan data baru, merubah data lama, menghapus data) buku, rak, kategori buku, penerbit, pengarang. c) Petugas bisa melakukan pencarian data buku berdasarkan judul, pengarang, penerbit. d) Sistem dapat menampilkan hasil pencarian buku oleh anggota yang berisi informasi kode buku, judul, pengarang, penerbit, alamat penerbit, nomor telpon penerbit, tahun terbit, jumlah buku yang tersedia di rak, kode rak penyimpanan</p>	-
6.	<p>Tidak adanya daftar status anggota yang aktif, pasif, registrasi</p>	<p>a) Terdapat suatu fitur untuk mencatat status aktif anggota. b) Anggota yang lebih dari 2 bulan tidak</p>	<p>a) Petugas dapat melakukan olah data anggota seperti : menginputkan data anggota baru</p>	-

		<p>melakukan peminjaman atau pengembalian buku akan berubah otomatis statusnya menjadi pasif.</p> <p>c) Anggota baru yang belum pernah meminjam akan memiliki status registrasi</p>	<p>dan system akan mengeset secara otomatis statusnya menjadi registrasi, melakukan perubahan data anggota lama dan menghapus data anggota yang sudah keluar (mahasiswa yang telah lulus dan wisuda).</p> <p>b) Sistem dapat melakukan perubahan status anggota menjadi pasif secara otomatis jika anggota lebih dari 2 bulan tidak melakukan peminjaman atau pengembalian buku.</p>	
7.	Dapat dilakukannya transaksi peminjaman judul buku yang sama oleh pemustaka dalam satu kali periode peminjaman	<p>a) Dibuat suatu fitur yang bisa melakukan seleksi buku yang dipinjam berdasarkan judul buku.</p> <p>b) Apabila history peminjaman ada yang sama persis dengan judul buku yang akan dipinjam maka system akan mengeluarkan list buku yang sama yang sudah dipinjam beserta kode pinjam, tanggal pinjam dan keterlambatan pinjaman</p>	<p>a) Petugas dapat melakukan olah data transaksi peminjaman dan pengembalian buku.</p> <p>b) Ketika anggota akan meminjam buku, system akan menampilkan form history data peminjaman untuk melakukan pengecekan buku yang belum dikembalikan.</p> <p>c) Sistem akan menolak penginputan transaksi peminjaman jika masih terdapat data buku yang belum dikembalikan</p> <p>d) Sistem akan menolak melakukan penginputan data transaksi peminjaman baru jika ditemukan kesesuaian jenis buku yang sama di history peminjaman yang statusnya masih belum dikembalikan.</p>	-
8.	Mahasiswa bisa meminjam lebih dari 3 buku dalam satu kali periode peminjaman	<p>a) Dibuat suatu fitur yang bisa melakukan seleksi jumlah buku yang bisa dipinjam oleh mahasiswa.</p> <p>b) Bila jumlah yang dipinjam lebih dari 3,</p>	<p>c) Petugas dapat melakukan olah data transaksi peminjaman dan pengembalian buku.</p> <p>d) Ketika anggota akan meminjam</p>	-

		maka sistem akan menolak transaksi peminjaman.	buku, system akan menampilkan form history data peminjaman untuk melakukan pengecekan buku yang belum dikembalikan. e) Sistem akan menolak melakukan penginputan data transaksi peminjaman baru jika diketahui ada anggota yang berstatus mahasiswa, meminjam buku lebih dari 3 buah	
9.	Apabila terjadi kesalahan ketika memasukkan data, data yang sudah terlanjur disimpan tidak dapat diedit.	Terdapat tombol edit di setiap form untuk mempermudah pengeditan ketika terjadi kesalahan dalam pencatatan datanya.	Sistem memiliki tombol edit di setiap formnya untuk melakukan perubahan data lama	-
10.	tidak adanya fasilitas pencarian berdasarkan nama, akibatnya mahasiswa yang lupa nimnya akan memakan waktu yang cukup lama dalam proses pencatatan transaksinya	Terdapat fitur pencarian berdasarkan nomor anggota, nim, nama anggota.	Sistem memiliki fitur untuk melakukan pencarian data anggota berdasarkan nomor anggota, nama anggota, PRODI anggota	-
11.	Laporan statistik pengunjung tidak dapat dilihat dalam periode harian dan mingguan.	Dibuatkan laporan dalam bentuk diagram batang yang bisa menjadi pembandingan jumlah pengunjung tiap bulannya dalam jangka waktu 1 tahun	a) Sistem dapat menampilkan laporan anggota registrasi, aktif maupun pasif di setiap bulannya dalam bentuk diagram batang b) Sistem dapat menampilkan laporan perkembangan anggota berdasarkan statusnya anggota registrasi, anggota aktif dan anggota pasif c) Sistem dapat menampilkan laporan perkembangan anggota berdasarkan PRODI nya	-
12.	Laporan statistik jumlah buku dan koleksi tidak dapat dilihat sehingga penambahan jumlah buku pertahun harus di hitung secara manual	a) Dibuatkan laporan dalam bentuk diagram batang yang bisa menjadi pembandingan jumlah buku yang dalam kondisi baik, rusak dan hilang.	a) Sistem dapat menampilkan laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam	-

		b) Laporan statistik bisa diseleksi untuk semua buku, per jenis buku, per tahun.	bentuk diagram batang b) Sistem dapat menampilkan laporan jumlah buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang	
13.	Di sistem informasi perpustakaan belum ada menu untuk pengadaan akibatnya penambahan koleksi lebih cenderung jumlah eksemplar bukan jumlah judul.	a) Dibuat suatu form yang bisa digunakan untuk memeriksa stok per judul buku. b) Judul buku yang stoknya kurang dari 2 dalam kurun waktu 1 bulan, akan berwarna merah.	a) Dibuat suatu form yang bisa digunakan untuk memeriksa stok per judul buku. b) Judul buku yang stoknya kurang dari 2 dalam kurun waktu 1 bulan, akan berwarna merah.	-
14.	Rekap buku rusak, hilang masih dilakukan secara manual dengan mencatatnya satu persatu. Akibatnya jumlah stok buku terkadang habis tanpa disadari oleh petugas.	c) Fitur ini bisa digunakan sebagai pertimbangan petugas untuk pengadaan buku baru.	c) Fitur ini bisa digunakan sebagai pertimbangan petugas untuk pengadaan buku baru.	
15.	Perhitungan denda bagi anggota yang terlambat meminjam masih dihitung secara manual menggunakan calculator sehingga rawan terjadi kesalahan hitung.	a) Perhitungan denda sudah dilakukan otomatis oleh aplikasi dengan mengurangi tanggal pinjam dengan tanggal kembali. b) Untuk menghilangkan tanggal merah, di setiap awal tahun admin menginputkan tanggal-tanggal yang merah terlebih dahulu.	a) Petugas bisa menginputkan tanggal-tanggal merah di awal tahun b) Sistem bisa menghitung secara otomatis keterlambatan peminjaman buku dengan mengurangi tanggal kembali dengan tanggal pinjamnya c) Sistem bisa menghitung denda secara otomatis dengan rumus : keterlambatan * denda perhari d) Sistem bisa menampilkan laporan rekapitulasi denda di setiap hari, tanggal maupun setiap bulannya e) Petugas bisa melihat laporan jumlah denda perhari, prtanggal maupun per bulan dengan login terlebih dahulu f) Ketika buku dikembalikan, sistem	-



			secara otomatis melakukan pengecekan keterlambatan buku	
16.	format laporan sulit dibaca sehingga menyulitkan petugas untuk mendapatkan informasi secara detail mengenai jumlah peningkatan anggota, transaksi dan buku	<ul style="list-style-type: none"> <li>a) Dibuatkan laporan dalam bentuk diagram batang yang bisa menjadi pembandingan jumlah buku yang dalam kondisi baik, rusak dan hilang.</li> <li>b) Statistik bisa diseleksi untuk semua buku, per jenis buku, per tahun.</li> </ul>	<ul style="list-style-type: none"> <li>a) Sistem dapat menampilkan laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</li> <li>b) Sistem dapat menampilkan laporan jumlah buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</li> <li>c) Petugas bisa melihat laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang dengan login terlebih dahulu</li> <li>d) Petugas bisa melihat laporan buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</li> </ul>	-
17.	Dengan jumlah petugas sebanyak 4 orang untuk melayani transaksi peminjaman dan pengembalian ternyata belum bisa mempercepat proses tersebut. Ini bisa dilihat pada response time system sebesar 7,84 menit.	<ul style="list-style-type: none"> <li>a) Dibuat suatu system yang mampu mengefektifkan kinerja petugas perpustakaan.</li> <li>b) Dengan asumsi 150 transaksi perhari dan jam kerja 450 menit perhari maka system baru yang diusulkan hanya memerlukan 2 orang petugas untuk melayani transaksi peminjaman dan pengembalian buku.</li> </ul>	Petugas dapat melakukan olah data buku rusak dan buku hilang	-
18.	Penundaan beberapa pekerjaan menyebabkan peningkatan biaya operasional perpustakaan	<ul style="list-style-type: none"> <li>c) Dengan 2 orang petugas saja maka waktu menganggur bisa diperkecil yaitu <math>450 - ((150/2) * 4) = 150</math> menit.</li> </ul>		

		d) 2 petugas lainnya ditugaskan untuk melakukan pendataan buku yg tersedia, rusak dan hilang		
19.	Waktu response time yang begitu besar menyebabkan beberapa pekerjaan olah data lain misalnya olah data buku baru, rusak banyak mengalami penundaan.	<p>a) Sistem yang diusulkan mampu menangani transaksi peminjaman atau pengembalian buku di setiap transaksinya dalam waktu 2 menit. Melakukan penambahan bandwidth.</p> <p>b) Menggunakan database My. SQL dalam penyimpanan datanya.</p> <p>c) Sistem dapat melakukan backup otomatis dalam setiap bulan.</p> <p>d) Dibuatkan laporan dalam bentuk diagam batang yang bisa menjadi pembanding jumlah pengunjung tiap bulannya dalam jangka waktu 1 tahun</p> <p>e) Dibuatkan laporan dalam bentuk diagram batang yang bisa menjadi pembanding jumlah buku yang dalam kondisi baik, rusak dan hilang.</p> <p>f) Statistik bisa diseleksi untuk semua buku, per jenis buku, per tahun.</p>	<p>a) Sistem dapat menampilkan laporan anggota registrasi, aktif maupun pasif di setiap bulannya dalam bentuk diagram batang</p> <p>b) Sistem dapat menampilkan laporan perkembangan anggota berdasarkan statusnya anggota regisrasi, anggota aktif dan anggota pasif</p> <p>c) Sistem dapat menampilkan laporan perkembangan anggota berdasarkan PRODI nya</p> <p>d) Sistem dapat menampilkan laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</p> <p>e) Sistem dapat menampilkan laporan jumlah buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</p> <p>f) Petugas bisa melihat laporan jumlah keseluruhan buku dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang dengan login terlebih dahulu</p> <p>g) Petugas bisa melihat laporan buku per PRODI yang dalam kondisi baik, rusak dan hilang di setiap bulannya dalam bentuk diagram batang</p>	<p>a) Penambahan jumlah bandwidth hingga 500 Mbps</p> <p>b) Menggunakan software MySQL untuk melakukan manipulasi data di database</p> <p>c) Sistem dapat melakukan backup data 1 bulan sekali</p> <p>d) Minimum hardware yang dipakai ....</p> <p>e) Sistem yang dibangun menggunakan software PHP</p>
20.	Informasi yang dibutuhkan petugas seringkali tidak bisa diberikan tepat waktu, akibatnya sering terjadi kesalahan perhitungan anggaran tahunan perpustakaan.	<p>a) Dibuatkan laporan dalam bentuk diagam batang yang bisa menjadi pembanding jumlah pengunjung tiap bulannya dalam jangka waktu 1 tahun.</p> <p>b) Laporan bisa menjadi pertimbangan petugas dalam menambah fasilitas perpustakaan</p> <p>c) Dibuatkan laporan dalam bentuk diagram batang yang bisa menjadi pembanding jumlah buku yang dalam kondisi baik, rusak dan hilang.</p> <p>d) Statistik bisa diseleksi untuk semua buku, per jenis buku, per tahun.</p>		

		e) Laporan ini bisa digunakan oleh pihak petugas untuk melakukan pengalokasian anggaran pembelian buku baru dalam 1 tahun		
21.	Belum adanya fasilitas pembatasan hak akses petugas dan petugas. Semua data dipegang oleh petugas, ketika petugas ingin mendapatkan laporan harus menunggu pencetakan dari petugas	<ul style="list-style-type: none"> <li>a) Sistem yang dibangun memiliki fasilitas login.</li> <li>b) Setiap pengguna aplikasi terkecuali anggota, jika ingin menggunakan harus login terlebih dahulu.</li> <li>c) Username dan password yang digunakan untuk login dibuat oleh seorang petugas dengan level admin.</li> <li>d) Setiap pengguna dibatasi pengaksesannya sesuai level yang didaftarkan.</li> </ul>	<ul style="list-style-type: none"> <li>a) Admin bisa melakukan olah data login (menambahkan login petugas, manajemen, merubah login petugas, manajemen dan menghapus login petugas serta manajemen)</li> <li>b) Untuk melakukan olah data login admin harus login (memasukkan username dan password) terlebih dahulu</li> </ul>	-
22.	Username dan password untuk masuk ke dalam system belum dilakukan pengenkripsian. Akibatnya petugas lain bisa melakukan perubahan username dan password petugas lainnya.	<ul style="list-style-type: none"> <li>a) Password yang diinputkan akan dilakukan enkripsi sebelum masuk ke dalam table.</li> <li>b) Enkripsi menggunakan algoritma SHA 3</li> </ul>	-	<ul style="list-style-type: none"> <li>a) Sistem secara otomatis akan melakukan enkripsi menggunakan algoritma SHS3 pada setiap password yang diinputkan.</li> <li>b) Data password yang tersimpan di table login isinya sudah dalam bentuk pengacakan oleh algoritma SHA 3</li> </ul>
23.	Petugas bisa melakukan penghapusan data tanpa adanya pengawasan dari pihak petugas. Hal ini memberikan peluang terjadinya manipulasi data yang bisa merugikan organisasi	<ul style="list-style-type: none"> <li>a) User dengan level petugas hanya bisa melakukan input data dan edit data saja.</li> <li>b) Apabila akan dilakukan penghapusan data harus memanggil admin.</li> </ul>	-	Petugas dan manajemen tidak memiliki wewenang untuk melakukan penghapusan data. Jika ada kesalahan dalam penginputan data dan harus dilakukan penghapusan, maka petugas dan manajemen harus menghubungi admin

				terlebih dahulu
24.	Transaksi penyimpanan data ke database sering mengalami eror. Penginputan data baru seringkali tidak tersimpan ke dalam table	Adanya vasilitas validasi berupa message yang akan memberitahukan kepada petugas jika data sudah benar-benar tersimpan di table	Sistem akan melakukan validasi dengan menampilkan sebuah message apabila data yang diinputkan sudah benar-benar tersimpan di table	-
35.	Belum adanya fasilitas backup membuat system semakin lama semakin lambat	a) Dibuat fasilitas backup yang akan membackup data perpustakaan di setiap bulannya. b) Backup dalam bentuk mysql.	-	Sistem akan melakukan backup secara otomatis di setiap bulannya
36.	a) Sumber daya manusia perpustakaan tidak memiliki tenaga yang berkompeten dalam bidang teknologi informasi dan komputer. b) Kendala teknis yang berkaitan dengan sistem (misalkan terjadi eror ketika ) dan ketergantungan dengan pihak kedua dalam memperbaiki system yang eror membuat kerja tidak efektif.	a) Dibangun suatu aplikasi yang tampilannya mempermudah user dalam menggunakan. b) Setiap fitur memiliki suatu penjelasan dan perintah yang mudah dimengerti pengguna	-	a) Form dan tampilan dibuat user friendly b) Semua fitur di aplikasi memiliki penjelasan dan perintah yang dimengerti oleh pengguna aplikasi c) Adanya pelatihan / Training mengenai penggunaan aplikasi pada petugas, manajemen dan admin
37.	Sistem pengkodean buku masih belum dilakukan pengaturan, akibatnya kode buku ditulis auto increment. Kondisi ini menyulitkan petugas untuk melakukan pengkategorian buku	Sistem baru sudah terdapat fitur untuk melakukan penggolongan buku berdasarkan PRODI. Masing-masing jenis buku sudah ditempatkan di rak khusus dengan aturan penomoran : ISBN/PRODI/TAHUN_TERBIT/BUKU_KE	a) Sistem dapat menggolongkan jenis buku berdasarkan PRODI, tahun terbit, dan kategori buku. b) Sistem dapat melakukan pengolahan data rak (menambah, mengedit dan menghapus data rak) serta melakukan pengolahan data jenis buku yang disimpan di masing-masing rak. c) Sistem dapat melakukan pengesetan secara otomatis untuk setiap kode buku baru yang diinputkan. Kode buku diatur dengan aturan penomoran : ISBN/PRODI/TAHUN_TERBIT/BUK	-

			U_KE	
38.	Sistem informasi perpustakaan yang ada sekarang kurang kompetibel dengan semua web browser yang ada. Web browser yang kompetibel dengan sistem ini hanya Internet Explorer saja.	Sistem yang dibangun harus compatible untuk dijalankan di browser apapun terutama firefox dan google chrome.	-	Sistem bisa dijalankan di web browser Mozilla firefox dan google chrome
39.	Sarana dan prasarana yang digunakan sudah cukup tua. Untuk spesifikasi komputer yang digunakan keluaran tahun 2005/2006 sudah tidak dapat memenuhi tuntutan pelayanan yang cepat kepada petugas.	<ul style="list-style-type: none"> <li>a) Perlu dilakukan upgrade PC untuk memaksimalkan penggunaan aplikasi.</li> <li>b) Diperlukan minimal 1 server dan 4 PC client.</li> </ul>	-	<ul style="list-style-type: none"> <li>a) Sistem yang dibangun memerlukan 1 computer server dan 4 computer client</li> <li>b) Computer server memiliki spesifikasi hardware :</li> <li>c) Processor ...., RAM ...., Harddisk ...., VGA ...., Monitor 20 inch, ....</li> <li>d) Computer client memiliki spesifikasi hardware : ....</li> <li>e) Diperlukan printer dengan spesifikasi ....</li> <li>f) Diperlukan scanner dengan spesifikasi ....</li> </ul>
40.	Sering tidak tercatatnya data yang diinputkan ke dalam suatu table di database aplikasi, membuat beberapa laporan menjadi tidak akurat, tidak konsisten karna terdapat data yang redundan di dalamnya. Misalnya : ketika transaksi peminjaman, data yang diinputkan tidak tersimpan ke table peminjaman. Akibatnya ketika terjadi transaksi pengembalian maka data peminjaman tidak dapat ditemukan.	<ul style="list-style-type: none"> <li>a) Adanya vasilitas validasi berupa message yang akan memberitahukan kepada petugas jika data yang diinputkan sudah benar-benar tersimpan di table.</li> <li>b) Sebaliknya untuk aktifitas penghapusan, system akan melakukan validasi apakah data tersebut benar-benar akan dihapus. Hal ini dilakukan untuk menghindari terjadinya kesalahan dalam penghapusan suatu data.</li> </ul>	Sistem akan melakukan validasi dengan menampilkan sebuah message apabila data yang ingin dihapus sudah benar-benar terhapus dari table	-

#### **4.2 Tugas**

Untuk kasus pengembangan sistem yang sedang anda kerjakan, tuliskan hasil analisa sistem yang berupa :

1. struktur kerja yang akan memberikan gambaran mengenai sistem yang sedang berjalan (gambarkan dengan flowchart sistem) dan harapan pemakai sistem pada sistem yang akan dibangun !
2. Penjabaran kelayakan teknis sistem yang akan dikembangkan (harus dijabarkan satu persatu mulai dari kelayakan teknis, kelayakan ekonomis, kelayakan legal dan kelayakan operasional) !
3. Buatlah identifikasi masalah, solusi kebutuhan yang diperlukan selama proses pengembangan sistem (buat dalam bentuk tabel) !Buatlah analisa kebutuhan fungsional dan nonfungsional untuk setiap solusi yang diusulkan di poin 3 !

---

## BAB V

### PERANCANGAN MENGGUNAKAN *DATA FLOW DIAGRAM* (DFD)

---

Pertemuan ke-8,9,10,11,12

Tujuan :

1. Mahasiswa memahami konsep *data flow diagram*
  2. Mahasiswa memahami tahapan-tahapan dalam pembuatan *data flow diagram*
  3. Mahasiswa mampu membuat perancangan DFD di suatu kasus
- 

#### 5.1 Pengertian DFD

Dalam analisis sistem, model digunakan untuk menampilkan atau menyajikan perancangan perangkat lunak. Pembagian model untuk melakukan perancangan perangkat lunak dapat dikelompokkan menjadi (Al Fatta, 2007):

1. Berorientasi Aliran Data (*Data Flow Oriented* atau *Functional Oriented*) . Sudut pandang analisis pada pendekatan ini difokuskan pada aspek fungsional dan behavioral (perilaku laku) sistem. Contoh diagram yang sering digunakan :
  - a. *Data Flow Diagram* (DFD) dan Kamus Data (*data dictionary*) untuk menggambarkan fungsi-fungsi dari sistem
  - b. *Entity-Relationship Diagram* (ERD) untuk menggambarkan data yang disimpan (data storage).
2. Berorientasi objek : Berbeda dengan pendekatan-pendekatan sebelumnya, pendekatan berorientasi objek memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Diagram yang digunakan adalah UML (*Unified Modelling Language*).

DFD diperkenalkan oleh Tom DeMarco serta Chris Gane dan Trish Sarson. DFD adalah model proses yang digunakan untuk menggambarkan aliran data melalui sebuah sistem dan tugas atau pengolahan yang dilakukan oleh sistem (Whitten, dkk (2004)). DFD merupakan suatu tampilan grafis yang memunculkan relasi/hubungan antara proses dan data beserta kamus data yang menjelaskan rincian data yang dipergunakan. Diagram ini juga digunakan untuk menggambarkan aliran data dalam sistem, sumber dan tujuan data, proses yang mengolah data tersebut, dan tempat penyimpanan datanya.

DFD merupakan salah satu teknik yang cukup penting dalam menganalisa sistem karena (Al Fatta, 2007):

1. Dapat mendefinisikan batasan sistem
2. Membantu memeriksa kebenaran dan kelengkapan aliran informasi
3. Merupakan dasar perancangan dengan memunculkan proses-proses pengolahan data.

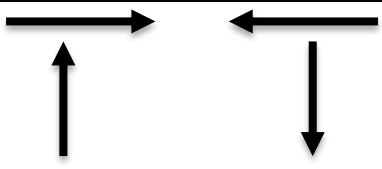
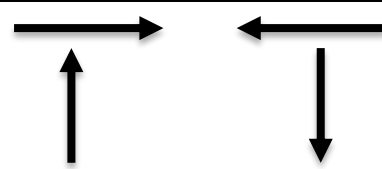
#### 5.2 Elemen dan Aturan Dalam Pembuatan DFD

Ada empat elemen yang membentuk suatu Data Flow Diagram, yaitu (Al Fatta, 2007):

1. Aliran data (*Data Flow*)
  - a. Penghubung antar proses yang merepresentasikan informasi yang dibutuhkan proses sebagai masukan atau informasi yang dihasilkan proses sebagai keluaran.
  - b. Aliran paket informasi dari satu bagian sistem ke bagian sistem lainnya. Umumnya mengalir antar proses, tetapi dapat juga mengalir keluar masuk dari ke file (data store) atau dari ke sumber tujuan data.

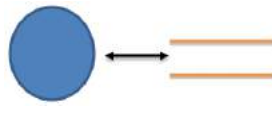
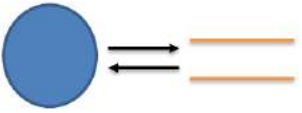
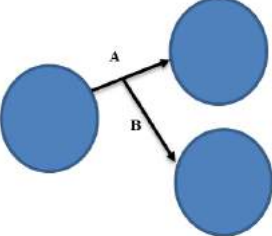
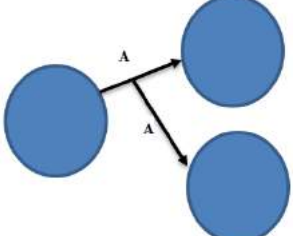
- c. Data yang dinyatakan dengan aliran data boleh datang dari beberapa dokumen, jadi tidak perlu dirinci menjadi dokumen-dokumen tersebut.
- d. Diberi nama sesuai dengan substansi isi dari paket informasi (bukan nama dokumen) yang mengalir.

**Tabel 2.** Simbol aliran data

Simbol	
DeMarco / Yourdon	Gane dan Sarson
	

Aturan dalam pembuatan aliran data adalah :

**Tabel 3.** Aturan pembuatan aliran data

Bagian	Aturan	Gambar Salah	Gambar Benar
Aliran data	Aliran data hanya boleh memiliki satu arah aliran data antara simbol yang satu dengan yang lainnya. Aliran dua arah bisa dimiliki antara proses dengan data store yang menunjukkan pembacaan data sebelum data diupdate, yang diidentifikasi dengan dua arah yang terpisah yang terjadi pada waktu yang berbeda atau penggambaran anak panahnya tidak boleh ganda.		
	Aliran data yang sama yang menuju beberapa proses, data store atau sumber/tujuan data berbeda boleh digambarkan bercabang.		



<p>Aliran data yang sama dari beberapa proses, data store atau sumber/tujuan data yang menuju suatu proses tertentu boleh digambarkan bercabang.</p>		
<p>Aliran data tidak boleh secara langsung mengalir ke dirinya sendiri (sirkuler). aliran data tersebut harus diproses minimal satu atau lebih proses yang akan menghasilkan beberapa aliran data yang lain dan kembali ke aliran data yang asli ke proses yang awal.</p>		
<p>Aliran data ke data store maksudnya mengupdate data baik berupa penghapusan data maupun perubahan data.</p>		
<p>Aliran data dari data store maksudnya proses mengambil atau membaca data dalam data store tersebut.</p>		
<p>Nama aliran data menggunakan kata benda. Beberapa aliran data dapat digunakan untuk satu anak panah asalkan kesemua data tersebut merupakan satu kesatuan paket data.</p>		

2. Proses

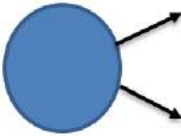
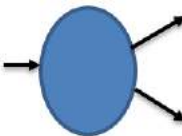
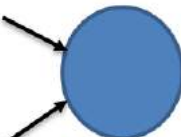
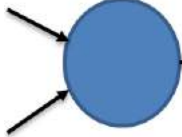
- a. Transformasi aliran data yang datang menjadi aliran data yang keluar.
- b. Transformasi bagaimana satu atau beberapa masukan diubah menjadi keluaran.
- c. Menjelaskan proses-proses transformasi data apa saja yang ada dalam sistem atau yang harus dikerjakan oleh sistem. Komponen-komponen fisik tidak dapat diidentifikasi sebagai proses.
- d. Diberi nama dan nomor yang akan dipergunakan untuk keperluan identifikasi. Nama yang diberikan harus dapat menjelaskan apa yang dilakukan oleh proses. Nama proses biasanya ditulis dalam kata kerja.

**Tabel 4.** Simbol proses

<b>Simbol</b>	
DeMarco / Yourdon	Gane dan Sarson

Aturan dalam pembuatan proses adalah :



**Tabel 5.** Aturan dalam pembuatan proses

Bagian	Aturan	Gambar Salah	Gambar Benar
Proses	Tidak ada proses yang hanya mempunyai data keluaran saja. Jika objek hanya mempunyai data keluaran, maka objek tersebut adalah sumber data.		
	Tidak boleh hanya mempunyai data masukan saja ( <i>black hole</i> ). Jika sebuah objek hanya mempunyai data masukan saja, maka objek tersebut adalah tujuan data.		
	Nama proses harus menggunakan kata kerja (misal olah data buku) atau nama yang dibendakan (misal pengolahan data buku)		

3. Penyimpanan Data (*Data Store*)

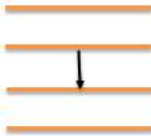
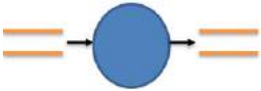
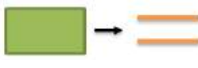
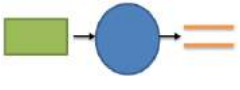
- a. Tempat penyimpanan data atau tempat data yang dirujuk oleh proses.
- b. Kumpulan paket data yang harus diingat oleh sistem dalam periode waktu tertentu.
- c. Pada akhir pembangunan sistem, data store biasanya diimplementasi sebagai file atau basis data.

**Tabel 6.** Simbol *data store*

Simbol	
DeMarco / Yourdon	Gane dan Sarson
	

Aturan dalam pembuatan *data store* adalah :

**Tabel 7.** Aturan dalam pembuatan *data store*



Bagian	Aturan	Gambar Salah	Gambar Benar
<i>Data Store</i>	Data tidak dapat mengalir langsung dari satu data store ke <i>data store</i> yang lain. Data harus berpindah melalui proses terlebih dahulu.		
	Data tidak dapat mengalir langsung dari sumber data ke <i>data store</i> . Data harus berpindah melalui proses terlebih dahulu.		
	Data tidak dapat mengalir langsung dari <i>data store</i> ke		

	tujuan data, data harus melalui proses terlebih dahulu.		
	Nama <i>data store</i> harus menggunakan kata benda (misal barang).		

4. Entitas Eksternal



- a. Menggambarkan entitas yang berinteraksi dengan sistem yang berada diluar ruang lingkup sistem (bukan yang menjalankan sistem tersebut) atau entitas yang berfungsi sebagai *producer/consumer* dari sistem (sumber atau tujuan data).
- b. Dapat berupa orang, unit organisasi, komputer eksternal, organisasi eksternal atau sistem lain. Operator yang memasukkan data dalam sistem termasuk entitas internal, karena ia bukan consumer/producer sistem (kecuali untuk ruang lingkup perangkat lunak tertentu).
- c. Antara terminator tidak boleh berkomunikasi langsung.
- d. Jumlah entitas/terminator yang terkait pada satu level akan muncul dalam jumlah yang sama untuk level lainnya

**Tabel 8.** Simbol *entitas eksternal*

Simbol	
DeMarco / Yourdon	Gane dan Sarson
	

Aturan dalam pembuatan *data store* adalah :

**Tabel 9.** Aturan dalam pembuatan *data store*

Bagian	Aturan	Gambar Salah	Gambar Benar
<i>Entitas Eksternal</i>	Data tidak dapat mengalir secara langsung dari sumber data ke tujuan data. Data harus melalui proses terlebih dahulu. Jikapun ada, hal tersebut tidak digambarkan dalam DFD.		
	Nama sumber/tujuan data harus menggunakan kata benda (misal petugas)		

**5.3 Tingkatan Dalam Pembuatan DFD**

Terdapat tingkatan dalam pembuatan DFD. Tingkatan itu dapat dituliskan dengan :

1. DFD level 0 (ada juga buku yang menyebutnya **diagram konteks**)  
Menggambarkan secara umum konteks yang terjadi dalam sistem antara dunia internal dan dunia eksternal yang berbatasan. Merupakan lapisan teratas terhadap sistem yang akan di bahas
2. DFD Level 1 ... s.d. Level n  
Merupakan gambaran rinci dari diagram konteks, makin tinggi levelnya maka akan makin dalam penjabaran rincian prosesnya

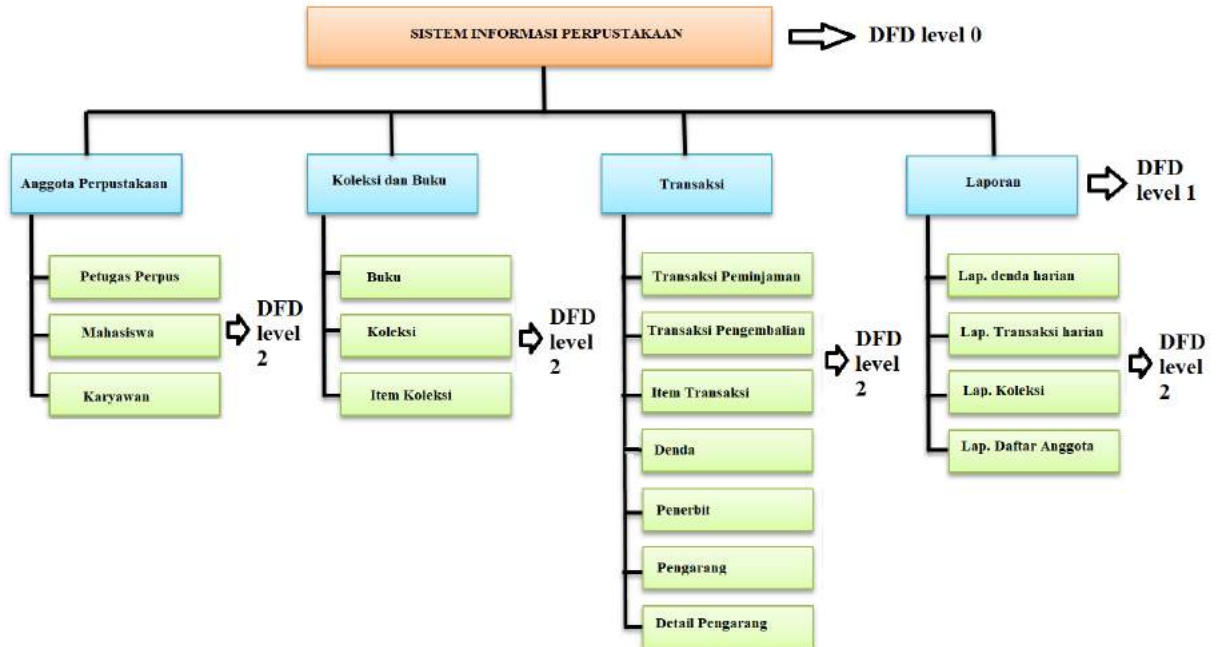
Secara singkat, langkah-langkah dalam pembuatan DFD adalah :

1. Membuat dekomposisi/struktur fungsional top-down suatu sistem.
2. Menentukan entitas luar yang terlibat dalam sistem tersebut. Pengidentifikasian entitas luar bisa diambil dari hasil survey di lapangan meliputi data siapa saja yang berinteraksi dengan sistem baik langsung maupun tidak dan asal serta tujuan data.
3. Menggambar diagram konteks, DFD level 1 dan DFD level 2.

Contoh kasus :

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.

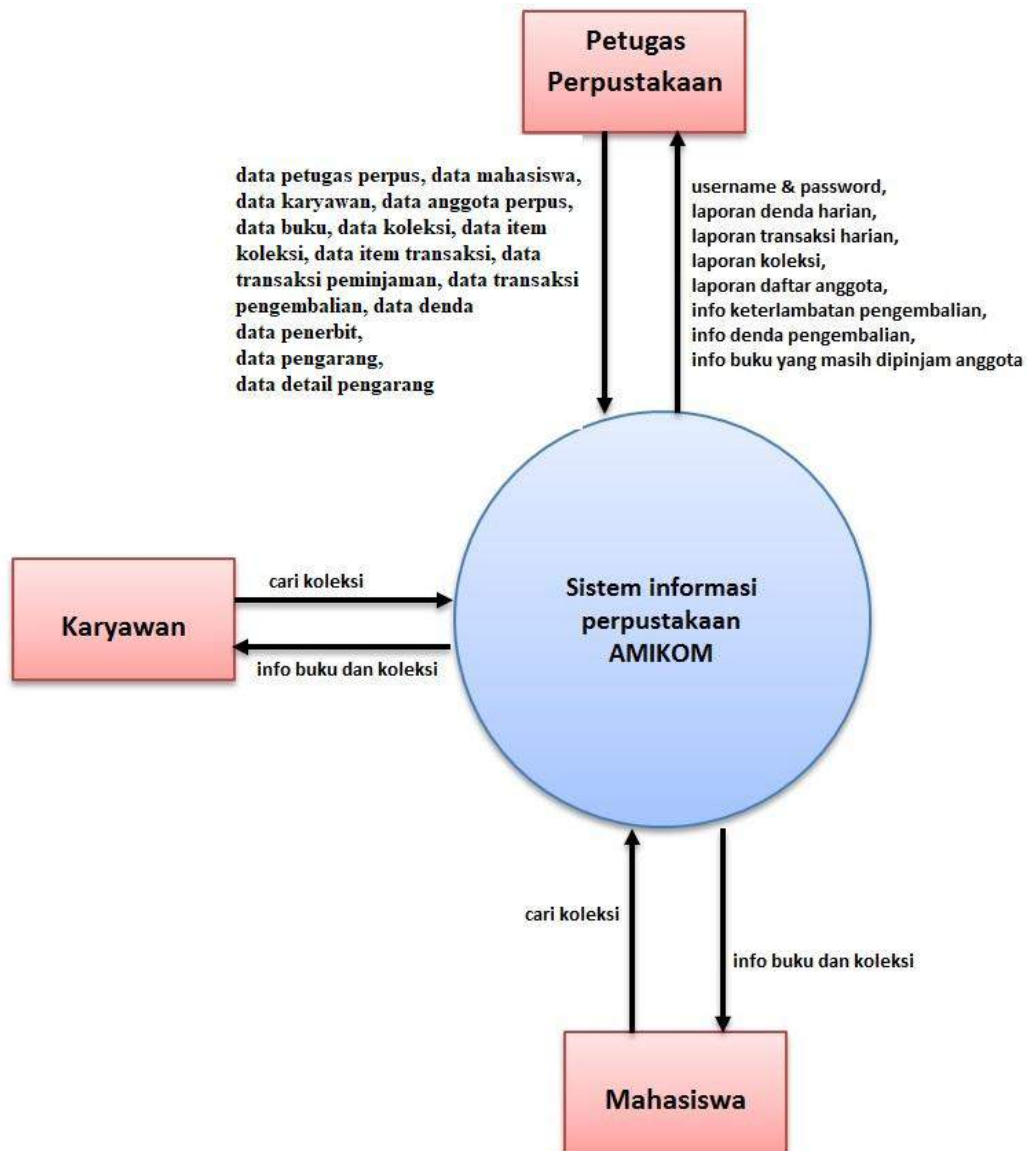
- Untuk membuat perancangan DFD dari kasus di atas, maka langkah-langkah yang dilakukan adalah :
1. Membuat dekomposisi/struktur fungsional top-down suatu sistem.



**Gambar 13.** Dekomposisi sistem informasi perpustakaan

2. Menentukan entitas luar yang terlibat dalam sistem tersebut.  
Entitas eksternal yang terlibat dalam penggunaan sistem informasi perpustakaan berdasarkan analisa yang telah dibuat ada 3 yaitu **petugas perpustakaan**, **karyawan** dan **mahasiswa**.

3. Membuat DFD level 0 (gambar 11)

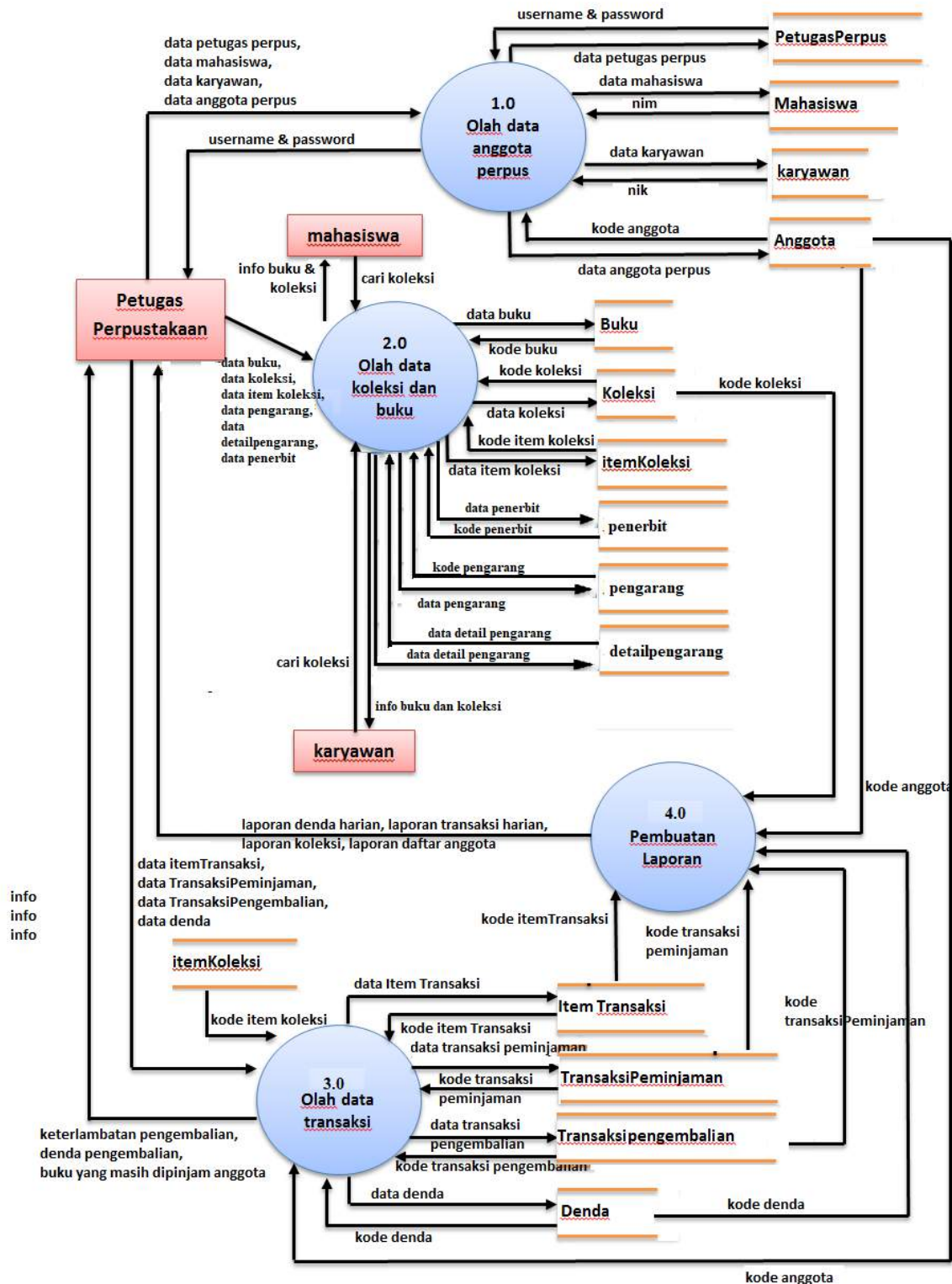


**Gambar 14.** DFD level 0 sistem informasi perpustakaan

Entitas eksternal yang pertama yaitu petugas perpustakaan. Petugas perpustakaan memberikan data petugas perpustakaan, data mahasiswa, data karyawan, data anggota perpustakaan, data buku, data koleksi, data item koleksi, data pengarang, data detail pengarang, data penerbit, data item transaksi, data transaksi peminjaman, data transaksi pengembalian, dan data denda ke sistem. sebagai timbal balik pemrosesan data yang masuk, petugas perpustakaan akan memperoleh data username dan password untuk login, informasi laporan denda harian, laporan transaksi harian, laporan koleksi, laporan daftar anggota, informasi keterlambatan pengembalian, informasi denda pengembalian dan informasi buku yang masih dipinjam oleh anggota.

Entitas eksternal lainnya adalah karyawan dan mahasiswa. Karyawan dan mahasiswa akan memasukkan kata kunci untuk pencarian koleksi ke dalam sistem dan memperoleh informasi buku dan koleksi yang dicari.

4. Menggambar diagram DFD level 1



Gambar 15. DFD level 1 sistem informasi perpustakaan

Petugas perpustakaan menginputkan data petugas perpustakaan ke sistem untuk kemudian disimpan dalam tabel petugas perpustakaan. Data ini oleh proses akan diolah untuk dihasilkan username dan password yang bisa digunakan oleh petugas perpus untuk login dan mengolah data di sistem informasi perpustakaan. Petugas perpustakaan juga menginputkan data mahasiswa, data karyawan dan data anggota perpus ke proses olah data anggota perpus. Oleh proses, data mahasiswa yang diinputkan akan disimpan dalam tabel mahasiswa sedangkan data karyawan akan disimpan dalam tabel karyawan dan data anggota perpus akan disimpan dalam tabel anggota. Data data ini akan diambil kembali untuk diolah di proses lain.

Di proses yang kedua yaitu olah data koleksi dan buku, petugas perpus akan menginputkan data buku, data koleksi dan data item koleksi. Data-data ini kemudian oleh proses akan disimpan secara berurutan ke dalam tabel buku, tabel koleksi dan tabel item koleksi.

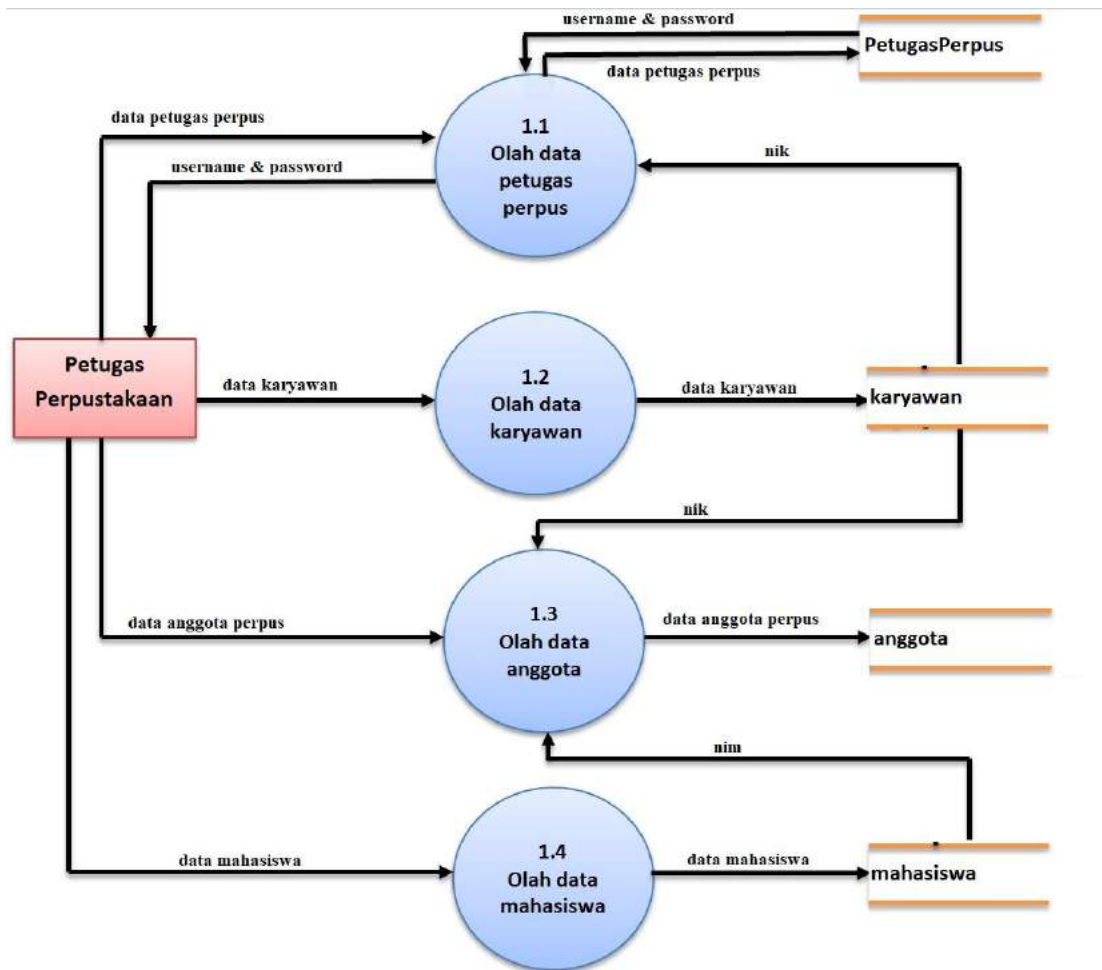
Ketika mahasiswa dan karyawan kebingungan untuk mencari daftar buku, maka mereka akan menginputkan info buku dan koleksi ke proses 2. Sebagai respon, proses 2 akan mencari daftar buku dan koleksi yang dicari ke dalam tabel buku dan tabel koleksi. Hasilnya, proses 2 akan menampilkan informasi buku dan koleksi yang dicari.

Ketika terjadi transaksi peminjaman dan pengembalian buku, petugas perpustakaan akan menginputkan data item transaksi, data transaksi peminjaman, data transaksi pengembalian dan data denda ke proses 3 olah data transaksi. Oleh proses 3, data ini secara berurutan akan disimpan dalam tabel itemtransaksi, tabel transaksipeminjaman, tabel transaksiPengembalian, dan tabel denda.

Dalam pembuatan laporan, proses 4 akan mengambil data di sejumlah tabel. Misalnya saja ketika akan membuat laporan anggota, maka proses 4 akan mengambil data di tabel anggota. Untuk membuat laporan denda harian, maka sistem akan mengambil data dari tabel denda. Untuk pembuatan laporan transaksi harian, sistem akan mengambil data dari tabel transaksiPeminjaman, transaksiPengembalian dan itemTransaksi. Sedangkan untuk membuat laporan koleksi, sistem akan mengambil data dari tabel buku dan tabel koleksi.



5. Menggambar DFD level 2 untuk proses 1 (olah data anggota perpustakaan)



Gambar 16. DFD level 2 proses 1 olah data anggota perpustakaan

DFD level 2 proses 1 dibuat untuk menggambarkan secara detail proses yang terjadi di proses 1 yaitu olah data anggota perpustakaan (gambar 15).

Proses pertama yang dilakukan adalah olah data petugas perpustakaan. Proses pertama ini akan menerima inputan dari petugas perpustakaan berupa data petugas perpustakaan untuk kemudian disimpan dalam tabel petugasPerpus. Sebaliknya proses ini akan memberikan username dan password yang bisa digunakan oleh petugas perpustakaan untuk login ke dalam sistem.

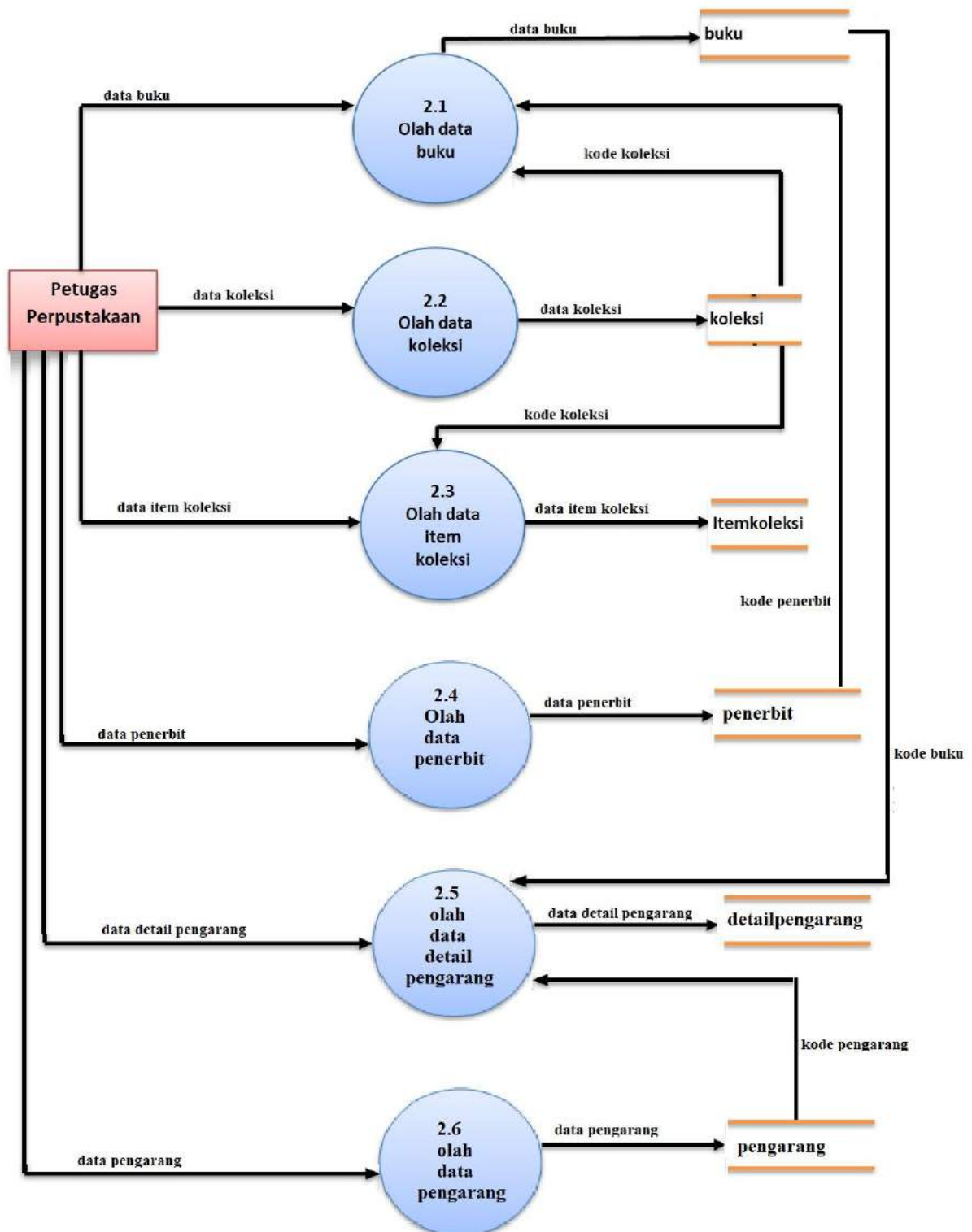
Petugas perpustakaan menginputkan data karyawan ke proses 1.2 (olah data karyawan) untuk kemudian data ini disimpan dalam tabel karyawan. nik yang ada di tabel karyawan akan diambil oleh proses 1.2 ketika memproses data petugas dan proses 1.3 ketika memproses data anggota.

Petugas perpustakaan menginputkan data mahasiswa ke proses 1.3 (olah data mahasiswa) untuk kemudian disimpan dalam tabel mahasiswa.

Ketika petugas perpustakaan menginputkan data anggota perpustakaan ke proses 1.3 (olah data anggota), proses akan mengambil nik di tabel karyawan dan nim di tabel mahasiswa yang sebelumnya telah diinputkan, untuk kemudian di proses bersama-sama dan disimpan dalam tabel anggota.



6. Menggambar DFD level 2 untuk proses 2 (olah data koleksi dan buku)



Gambar 17. DFD level 2 proses 2 olah data koleksi dan buku

DFD level 2 proses 2 dibuat untuk menggambarkan secara detail proses yang terjadi di proses 2 (gambar 15) yaitu olah data koleksi dan buku.

Proses pertama yang dilakukan adalah olah data koleksi (proses 2.2). Pada proses ini, petugas perpustakaan akan menginputkan data koleksi untuk kemudian oleh proses 2.2 akan diinputkan ke dalam tabel koleksi.

Selanjutnya, petugas perpustakaan juga akan menginputkan data buku ke dalam proses olah data buku. Data buku dan kode koleksi, kode penerbit yang diambil dari tabel koleksi, akan bersama-sama diolah oleh proses 2.1 dan disimpan dalam tabel buku.

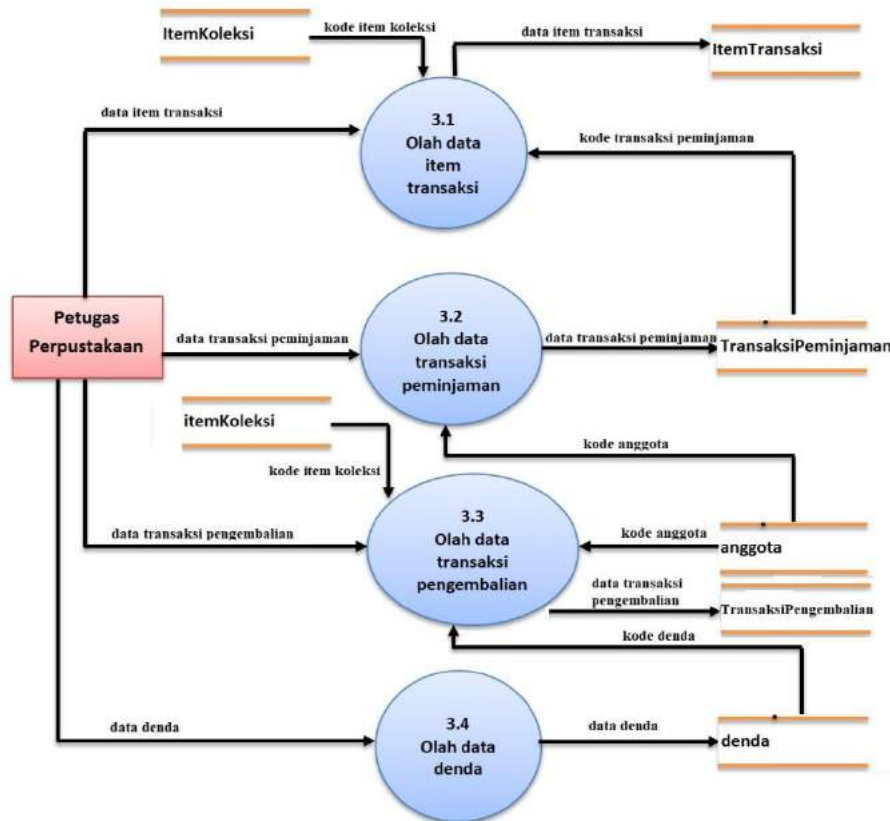
Pada proses 2.3 olah data item koleksi, petugas perpustakaan akan menginputkan data item koleksi ke proses 2.3. Data item koleksi dan kode koleksi yang diambil dari tabel koleksi akan diolah bersama-sama oleh proses 2.3 dan disimpan dalam tabel itemkoleksi.

Proses 2.4 olah data penerbit, petugas perpustakaan akan menginputkan data penerbit dan hasilnya akan disimpan dalam tabel penerbit.

Pada proses 2.6 olah data pengarang, petugas perpustakaan akan menginputkan data pengarang dan hasilnya akan disimpan dalam tabel pengarang.

Proses 2.7 olah data detail pengarang, petugas perpustakaan akan menginputkan data detail pengarang, kode buku yang diambil dari tabel buku dan kode pengarang yang diambil dari tabel pengarang untuk bersama-sama di proses oleh proses 2.7 dan hasilnya akan disimpan dalam tabel detail pengarang.

7. Menggambar DFD level 2 untuk proses 3 (olah data transaksi)



Gambar 18. DFD level 2 proses 3 olah data transaksi

DFD level 2 proses 3 dibuat untuk menggambarkan secara detail proses yang terjadi di proses 3 (gambar 15) yaitu olah data transaksi.

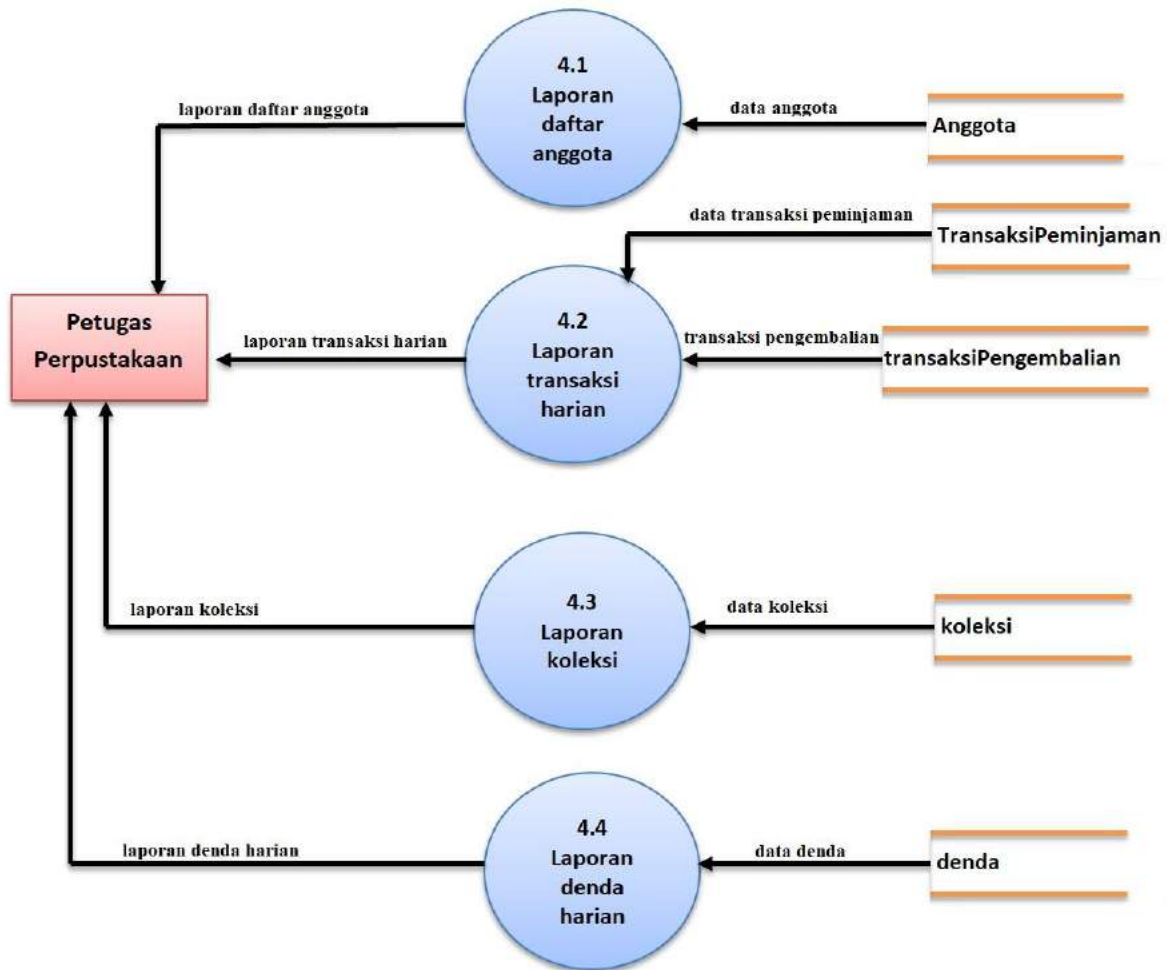
Proses pertama yang dilakukan adalah olah data transaksi peminjaman (proses 2.2). Proses ini dilakukan ketika ada anggota perpustakaan yang meminjam buku. Di proses 3.2 ini, petugas perpustakaan akan menginputkan data transaksi peminjaman. Data ini bersama sama dengan kode anggota yang diambil dari tabel tabel anggota akan diproses oleh proses 3.2 untuk kemudian disimpan dalam tabel transaksiPeminjaman.

Buku yang dipinjam, akan diproses oleh proses 3.1 (olah data item transaksi). Pada proses 3.1, petugas perpustakaan akan menginputkan data data item transaksi ke proses 3.1. data ini bersama sama dengan kode transaksi peminjaman yang diambil dari tabel transaksipeminjaman, kode item koleksi yang diambil dari tabel itemkoleksi akan diproses oleh proses 3.1 dan disimpan dalam tabel itemTransaksi.

Buku yang dikembalikan akan di proses di proses 3.3 (olah data transaksi pengembalian). Petugas perpustakaan akan menginputkan data transaksi pengembalian buku untuk kemudian bersama sama dengan kode item koleksi yang diambil dari tabel itemKoleksi, kode anggota dari tabel anggota dan kode denda dari tabel denda, akan diproses dan disimpan dalam tabel transaksiPengembalian.

Data denda sendiri akan diproses oleh proses 3.4 (olah data denda) dan hasil pemrosesannya disimpan dalam tabel denda.

8. Menggambar DFD level 2 untuk proses 4 (pembuatan laporan)



Gambar 19. DFD level 2 proses 3 olah data transaksi

Pembuatan laporan daftar anggota dilakukan dengan mengambil data anggota dari tabel anggota yang telah diinputkan sebelumnya di proses 1 (gambar 15). Sedangkan untuk laporan transaksi harian, data diambil dari data transaksi peminjaman yang sebelumnya disimpan di tabel TransaksiPeminjaman dan data transaksi pengembalian yang disimpan dalam tabel transaksiPengembalian. Untuk laporan koleksi sendiri data diambil dari tabel koleksi. Untuk laporan denda harian, data diambil dari tabel denda. Laporan daftar anggota, laporan transaksi harian, laporan koleksi dan laporan denda harian akan dipakai petugas perpustakaan sebagai bahan evaluasi sistem yang berjalan.

#### 5.4 Tugas

1. Buatlah gambar dekomposisi/struktur fungsional top-down sistem dari tugas anda !
2. Buatlah DFD level 0 dari tugas tersebut !
3. Buatlah DFD level 1 dari tugas tersebut !
4. Buatlah DFD level 2 dari tugas tersebut !

---

## BAB VI

### PERANCANGAN *DATABASE* MENGGUNAKAN *ENTITY RELATIONSHIP DIAGRAM (ERD)*

---

Pertemuan ke-13,14,15

Tujuan :

1. Mahasiswa memahami konsep *entity relationship diagram*
  2. Mahasiswa memahami tahapan-tahapan dalam pembuatan *entity relationship diagram*
  3. Mahasiswa mampu mengimplementasikan ERD pada suatu kasus
- 

#### 6.1 Definisi ERD

Pemodelan sistem database dapat dilakukan melalui pendekatan perancangan secara konseptual yaitu *Entity Relationship Diagram (ERD atau Er Diagram)*. *Entity Relationship Diagram (ERD)* adalah diagram yang menggambarkan keterhubungan antar entitas. ER Diagram menggambarkan tipe objek mengenai data itu di manajemen, serta relasi antara objek tersebut. ER Diagram digunakan oleh seorang analis sistem dalam merancang database. ER Model dibuat berdasarkan pengamatan saat pengumpulan kebutuhan sistem (*requirement system*) yang terdiri atas entitas dan relasi antar entitas-entitas tersebut.

Ada beberapa notasi yang digunakan untuk membuat ER Diagram. Misal. notasi Chen, Martin, El Masri dan Korth, akan tetapi pada umumnya adalah sama. Perbedaannya adalah pada pemilihan simbol-simbol yang digunakan. Pada materi komponen ER diagram ini, digunakan notasi El Masri karena lebih umum banyak digunakan dan mudah dibaca dan dimengerti.

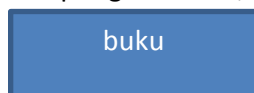
#### 6.2 Komponen ERD

Komponen di ER Diagram :

##### 1. Entity/Entitas

Entitas adalah sebuah obyek yang ada (exist) dan dapat dibedakan dengan obyek yang lain. Entitas ada yang bersifat konkrit, seperti: orang (pegawai, mahasiswa, dosen, dll), buku, perusahaan; dan ada yang bersifat abstrak, seperti: buku, petugas perpustakaan, koleksi, transaksi peminjaman, transaksi pengembalian, denda dan sebagainya.

**Simbol entitas :**



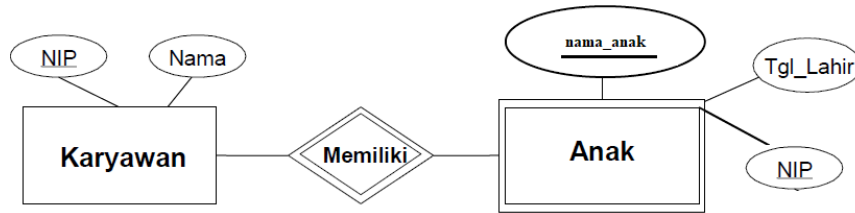
Beberapa entitas kemungkinan tidak memiliki atribut kunci sendiri (istilah atribut dijelaskan di bawah), entitas demikian disebut Entitas Lemah (*Weak Entity*). Entitas Lemah (*Weak Entity*) adalah entitas yang keberadaannya sangat bergantung dengan entitas lain.

Ciri dari *weak entity* :

- a. Tidak memiliki *Key Attribute* sendiri.
- b. Entitas tempat bergantung disebut *Identifying Owner / Owner*.
- c. Entitas lemah tidak memiliki *identifier*-nya sendiri.

Atribut entitas lemah berperan sebagai *Partial Identifier* (identifikasi yang berfungsi secara sebagian).

**Contoh weak entity :**



Implementasi dari *weak entity* di atas adalah :

Nama tabel : karyawan	
Nip	nama
190802256	Budi
190507030	Ayu
....	....

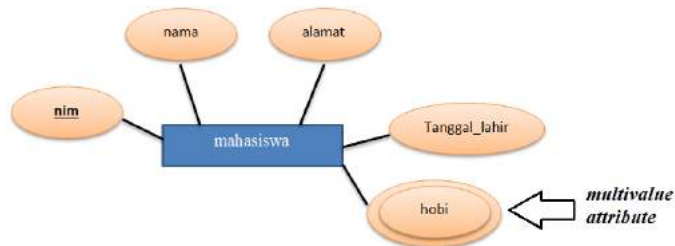
Nama tabel : anak		
nip	Nama_anak	Tanggal_lahir
12547	Agus	21/11/2018
12548	Fitri	2/4/2010
....	....	....

2. Atribut

Setiap entitas memiliki atribut sebagai keterangan dari entitas, misalnya entitas mahasiswa, yang memiliki atribut: nim, nama dan alamat.

Macam atribut antara lain :

- a) *Multivalue attribute* (atribut bernilai banyak). atribut dari entitas yang memiliki lebih dari satu nilai. Contoh: Atribut hobi.



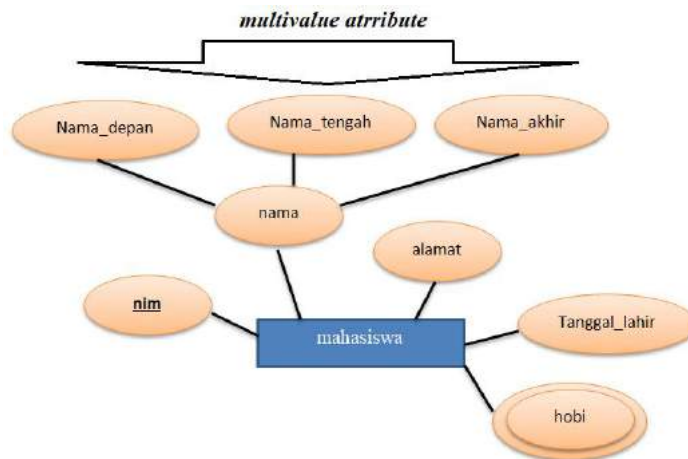
Implementasi dari entitas mahasiswa di atas adalah :

Nama tabel : mahasiswa				
nim	nama	alamat	tanggal_lahir	hobi
16.02.2564	Budi	Jl. Siliwangi rt.3 rw.21 no. 8	21/11/2000	Membaca, melukis, browsing, renang
16.02.2565	Ayu	Jl. Kuta rt.4 rw.2 no. 20	5/6/1999	Browsing, menulis
....	....	....	....	....

b) *Composite attribute* (atribut komposit)

Atribut komposit adalah atribut yang terdiri dari beberapa atribut yang lebih mendasar/lebih kecil lagi.

Contoh : atribut nama

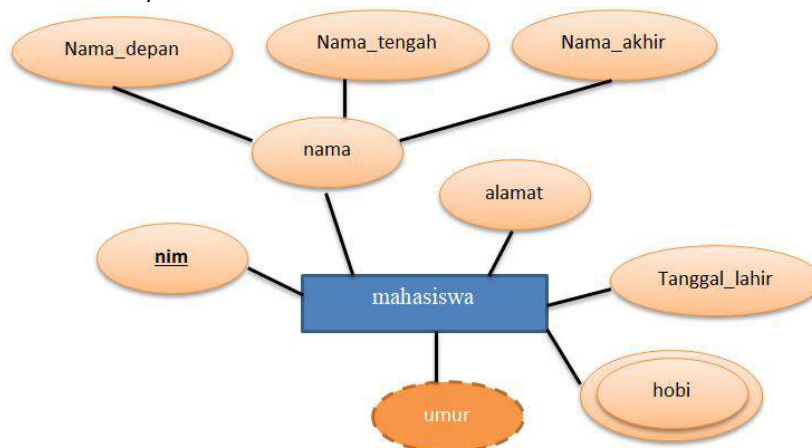


Implementasi dari entitas mahasiswa di atas adalah :

Nama tabel : mahasiswa						
nim	nama_depan	nama_tengah	nama_belakang	alamat	tanggal_lahir	hobi
16.02.2564	Budi	Prakoso	Wijaya	Jl. Siliwangi rt.3 rw.21 no. 8	21/11/2000	Membaca, melukis, browsing, renang
16.02.2565	Ayu	Kurnia	Cipta	Jl. Kuta rt.4 rw.2 no. 20	5/6/1999	Browsing, menulis
....	....			....	....	....

c) *Derived attribute* adalah atribut yang nilainya dapat diisi atau diturunkan dari perhitungan atau algoritma tertentu.

Contoh : atribut umur tidak diinputkan oleh user, namun dengan mengurangi secara otomatis tahun sekarang dengan tahun lahir (contoh untuk Budi, umur dihitung dengan cara 2018 – 2000 = 18 tahun).



Implementasi dari entitas mahasiswa di atas adalah :

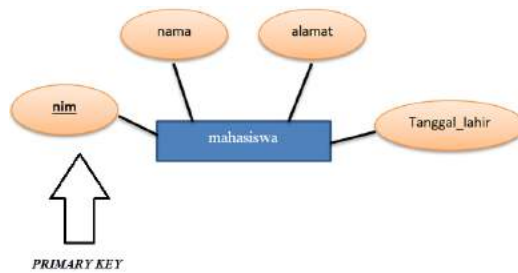
Nama tabel : mahasiswa							
nim	nama_depan	nama_tengah	nama_belakang	alamat	tanggal_lahir	hobi	umur
16.02.2564	Budi	Prakoso	Wijaya	Jl. Siliwangi rt.3 rw.21 no. 8	21/11/2000	Membaca, melukis, browsing, renang	18
16.02.2565	Ayu	Kurnia	Cipta	Jl. Kuta rt.4 rw.2 no. 20	5/6/1999	Browsing, menulis	19
....	....	....	....	....	....	....	....

3. Jenis kunci (key)

- a. *Primary key* : atribut yang bersifat unik yang bisa membedakan antara data yang satu (field) dalam satu entitas. Atribut yang menjadi *primary key* harus diberi garis bawah pada namanya.

Contoh :

atribut pada entitas mahasiswa ada 4 yaitu nim, nama, alamat, dan tanggal\_lahir. Dari sejumlah atribut tersebut, maka *primary key* yang bisa membedakan antara field satu dengan field lainnya pada entitas mahasiswa adalah nim. Nilai nim antara satu mahasiswa dengan mahasiswa lainnya tidak boleh sama.



Implementasi dari entitas mahasiswa di atas adalah :

Nama tabel : mahasiswa			
nim	nama	alamat	tanggal_lahir
16.02.2564	Budi	Jl. Siliwangi rt.3 rw.21 no. 8	21/11/2000
16.02.2565	Ayu	Jl. Kuta rt.4 rw.2 no. 20	5/6/1999
....	....	....	....

- b. *Foreign key* (kunci tamu). *Foreign key* adalah kunci yang digunakan untuk merelasikan/menghubungkan dua tabel. Entitas yang dimana atribut tersebut bertindak sebagai *foreign key* disebut tabel anak, dan entitas dimana atribut tersebut menjadi *primary key* disebut tabel referensi atau induk.

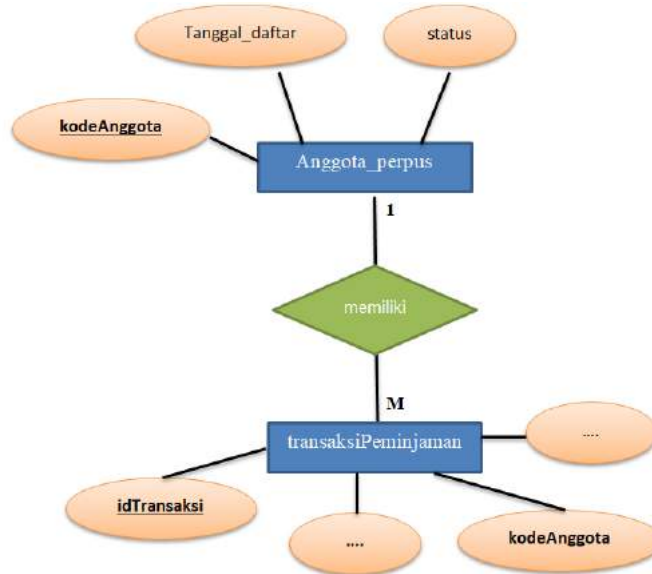
Contoh :

Atribut pada entitas mahasiswa ada 4 yaitu nim, nama, alamat, dan tanggal\_lahir.

Atribut pada entitas anggota\_perpus ada 4 juga yaitu kodeAnggota, nim, status, dan tanggal\_daftar.

Atribut yang sama pada entitas mahasiswa dan anggota\_perpus adalah nim. Nim di mahasiswa adalah *primary key*, sedangkan nim di entitas anggota\_perpus adalah *foreign key*.





Implementasi dari contoh di atas adalah :

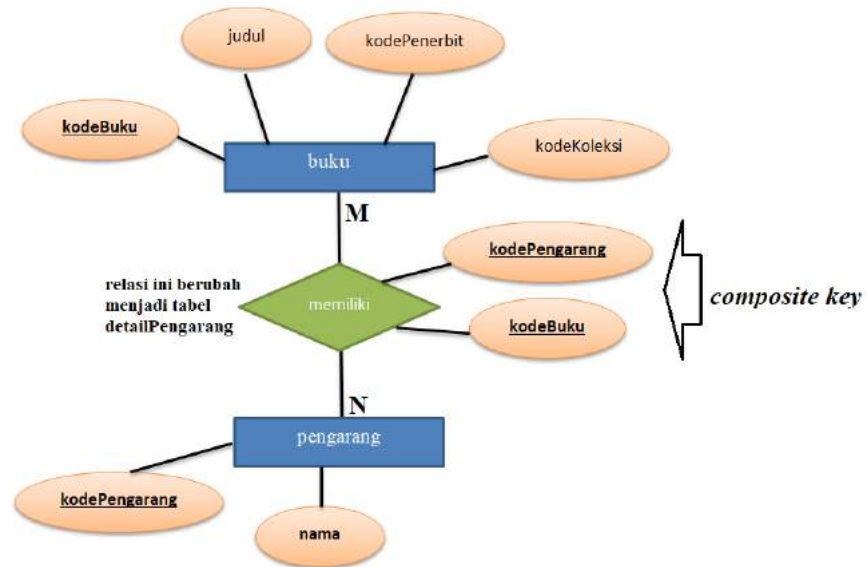
Nama tabel : anggota_perpus			
kodeAnggota	nim	Tanggal_daftar	Status
12547	16.02.2564	21/11/2018	Aktif
12548	16.02.2565	21/11/2018	Non aktif
....	....	....	....

Nama tabel : transaksiPeminjaman			
idTransaksi	kodeAnggota	....	....
Tr-0080	12547	....	....
Tr-0081	12524	....	....
Tr-0082	12547	....	....
Tr-0083	12520	....	....
....	....	....	....

c. *Composite key/kunci gabungan*

*Composite key* adalah kombinasi dari dua atau lebih kolom dalam entitas yang dapat digunakan untuk mengidentifikasi setiap baris dalam entitas secara unik. Keunikan hanya dijamin ketika atribut digabungkan; bila diambil secara individual atribut tidak menjamin keunikan. Biasanya muncul ketika ada relasi *many to many* (N to M). untuk relasi/kardinalitas silahkan lihat penjelasan di bawah.

Contoh : kardinalitas antara entitas buku dan entitas pengarang adalah *many to many*. Kardinalitas *many to many* ini menghasilkan entitas detailPengarang yang *key* nya bersifat *composite*.



Implementasi dari contoh di atas adalah :

Nama tabel : buku			
Kode_buku	judul	kodePenerbit	kodeKoleksi
1144	Analisa dan Desain Sistem	BI-0004	Computer-01
1145	RPL	Andy-0018	Computer-01
....	....	....	....

Nama tabel : detailPengarang	
kodePengarang	Kode_buku
P0001	1144
P0002	1145
P0003	1145
....	....

Nama tabel : Pengarang	
kodePengarang	nama
P0001	Rinaldi Munir
P0002	Iqbal Hasan
P0003	Misbahuddin
....	....

#### 4. Kardinalitas

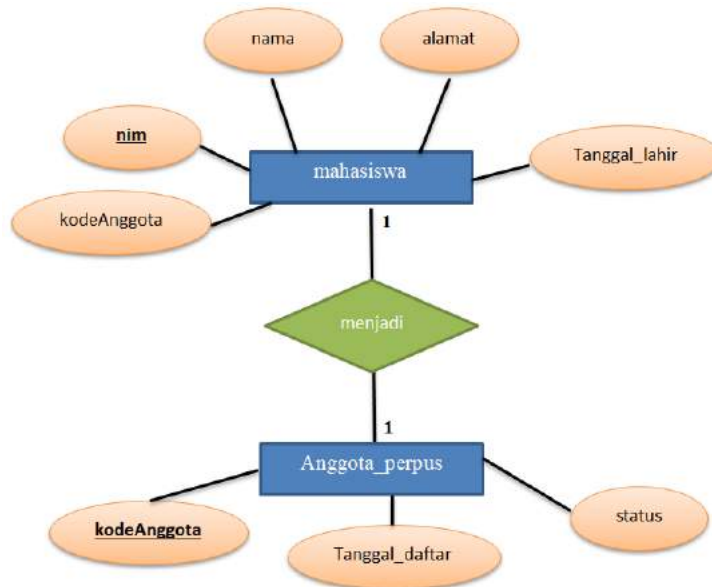
Menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

Jenis kardinalitas :

##### a) 1 : 1 (One-To-One)

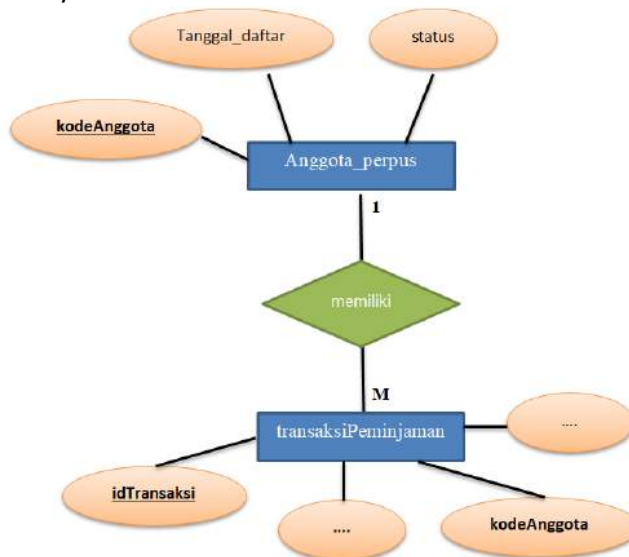
Sebuah entity A diasosiasikan pada sebuah entity B, dan sebuah entity B diasosiasikan dengan paling banyak sebuah entity A.

Contoh : seorang mahasiswa hanya bisa terdaftar satu kali menjadi anggota perpustakaan



b) 1 : N (*One-To-Many*)

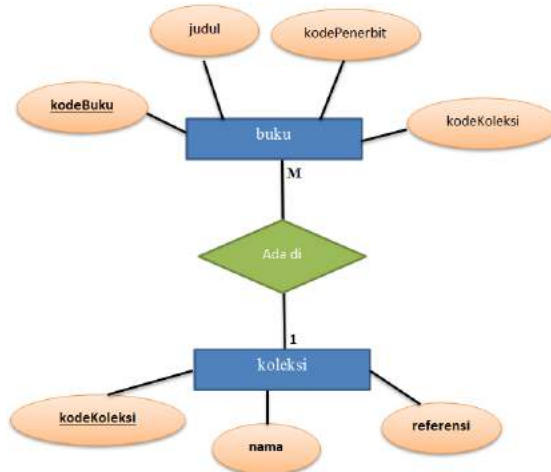
Sebuah entity A diasosiasikan dengan sejumlah entity B, tetapi entity B dapat diasosiasikan paling banyak satu entity A.



Dibaca : 1 anggota perpustakaan bisa melakukan banyak transaksi peminjaman

c) N : 1 (*Many-To-One*)

Suatu entity A dapat diasosiasikan dengan paling banyak sebuah entity B, tetapi entity B dapat diasosiasikan dengan sejumlah entity di A  
 Contoh ;

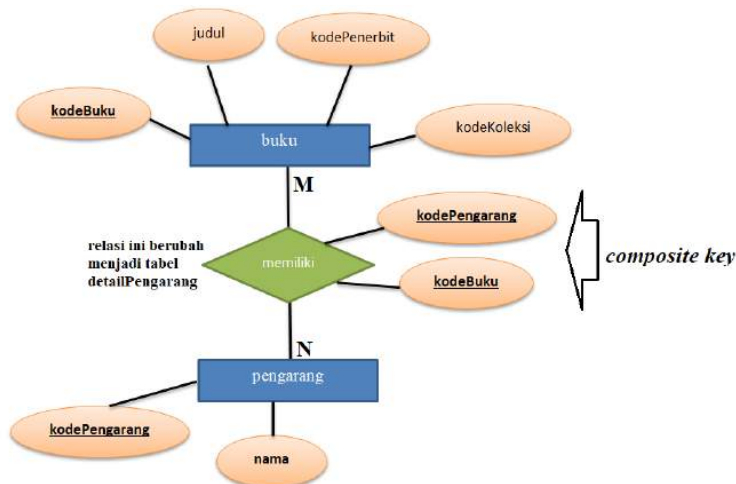


Dibaca : Banyak buku ada di 1 koleksi

d) N : M (*Many-To-Many*)

Suatu entity A dapat diasosiasikan dengan banyak entity B, dan entity B dapat diasosiasikan dengan banyak entity di A

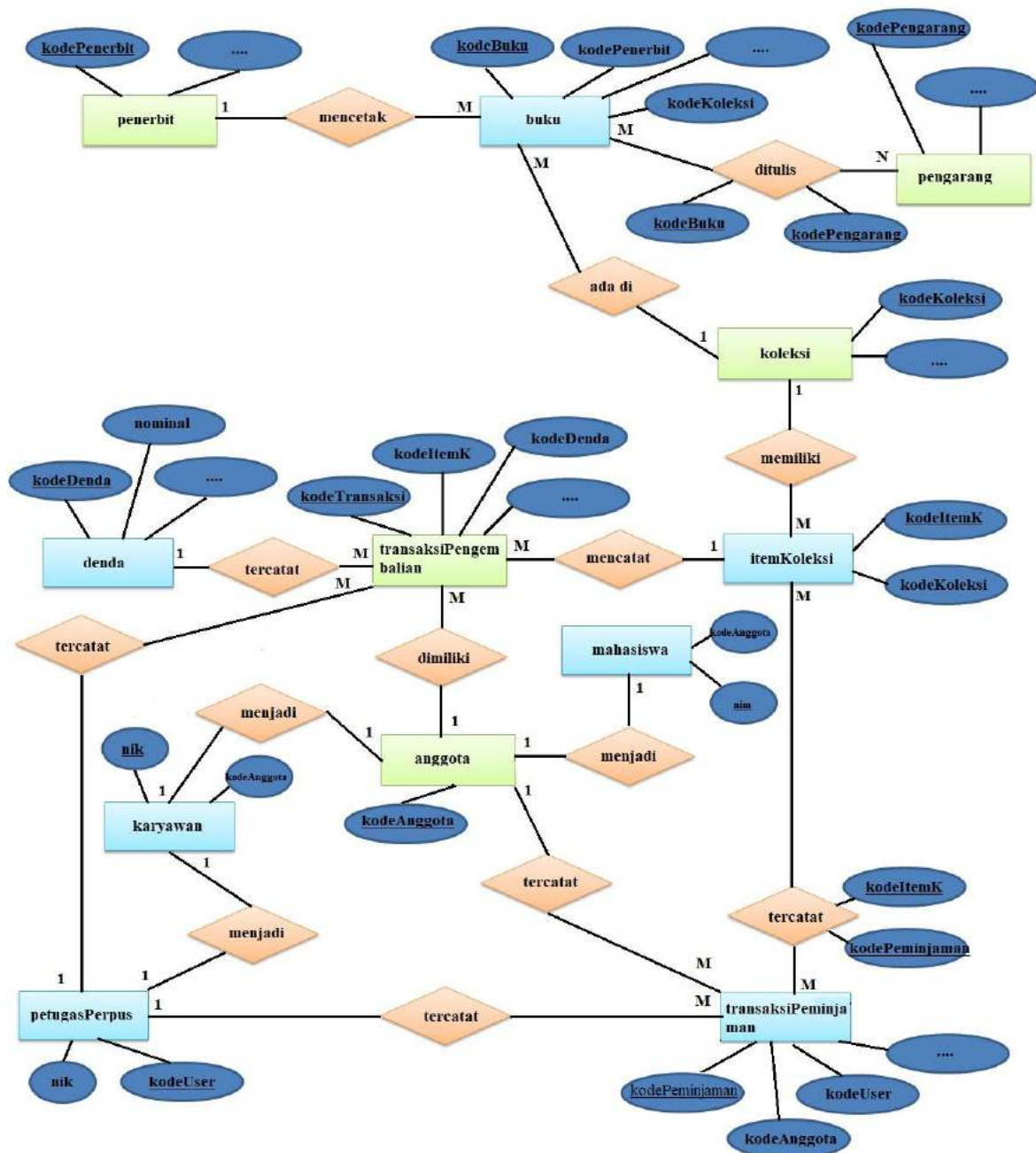
Contoh :



1 buku memiliki banyak pengarang dan 1 pengarang memiliki banyak buku.

Contoh kasus :

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda. Implementasi *entity relationship diagram* dari kasus di atas dapat dilihat pada gambar 17.



Gambar 20. Entity relationship diagram sistem informasi perpustakaan

### 6.3 Tugas

1. Buatlah implementasi tabel dari gambar 20 Entity relationship diagram sistem informasi perpustakaan !
2. Buatlah ERD dan implementasi tabel dari kasus pengembangan sistem yang sedang kelompok kalian buat !

## BAB VII

### PERANCANGAN SISTEM MENGGUNAKAN UNIFED MODELLING LANGUAGE (UML)

Pertemuan ke-16,17,18,19,20,21,22,23

Tujuan :

1. Mahasiswa memahami konsep UML
2. Mahasiswa memahami tahapan-tahapan dalam pembuatan UML
3. Mahasiswa mampu mengimplementasikan UML pada suatu kasus

#### 7.1 Konsep dasar OOP (Object Oriented Programming)

##### 7.1.1 Object

Pada dunia perangkat lunak, sebuah obyek adalah sebuah komponen perangkat lunak yang strukturnya mirip dengan obyek pada dunia nyata. Disekeliling kita ada banyak contoh dari objek, seperti meja, pena, kursi, televisi dan masih banyak lagi. Yang dimiliki oleh objek antar lain [14]:

1. *State* : pada implementasinya berupa *field* dari kelas
2. *Behaviour* : implementasinya menjadi *method* dari kelas

##### 7.1.2 Kelas (Class)

Kelas adalah keluarga dari Objek. Apabila beberapa objek memiliki struktur yang sama, perilaku dan arti secara umum, maka bisa dikelompokkan menjadi sebuah kelas. Yang kemudian dapat digunakan untuk menciptakan sebuah objek yang memiliki keseragaman. Setiap kelas memberikan skema yang umum untuk satu atau lebih objek, dimana kelas dapat menjadi template dari satu atau banyak objek [14].



Gambar 21. Implementasi dari kelas

### 7.1.3 Enkapsulasi

Ini adalah salah satu fitur penting di OOP. *Enkapsulation* atau enkapsulasi adalah suatu cara untuk menyembunyikan detail dari suatu proses atau cara untuk memberikan akses tidak langsung ke data. Dengan enkapsulasi, data yang ada dalam sistem lebih aman dan konsisten. Enkapsulasi juga akan membuat detail proses yang rumit tersembunyi sehingga para pengguna suatu Class atau obyek akan menemukan suatu kemudahan dimana. Kemudahan itu didapat karena mereka tidak perlu mengetahui detail suatu proses untuk dapat melakukan suatu proses, cukup dengan nama Method yang dibutuhkan [14].

Contohnya diketahui bahwa kelas Database memiliki 4 buah method yaitu

1. connect
2. dml
3. getGeneratedKey
4. getRS

Ketika anda hendak melakukan *Query* ke database misalnya untuk mengambil data Mahasiswa yang menjadi anggota perpustakaan maka cukup dengan perintah seperti dibawah ini :

```
@Override
public Mahasiswa[] findAll() {
    // TODO Auto-generated method stub
    ss="select a.idAnggota,nim,nama,tmplahir,tgllahir,tahunmasuk ";
    ss+="from mahasiswa m ";
    ss+="inner join anggota a on m.idanggota=a.idanggota";
    if(database.dml(ss, true)){
        rs=database.getRs();
        List<Mahasiswa> lstMahasiswa=new ArrayList<Mahasiswa>();
        try {
            while(rs.next()){
                Mahasiswa mahasiswa=new Mahasiswa();
                mahasiswa.setId(rs.getInt("idAnggota"));
                mahasiswa.setNim(rs.getString("nim"));
                mahasiswa.setNama(rs.getString("nama"));
                mahasiswa.setTmpLahir(rs.getString("tmplahir"));
                mahasiswa.setTglLahir(rs.getDate("tgllahir"));
                mahasiswa.setTahunMasuk(rs.getInt("tahunmasuk"));
                lstMahasiswa.add(mahasiswa);
            }
            rs.close();
            return lstMahasiswa.toArray(
                new Mahasiswa[lstMahasiswa.size()]);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            return null;
        }
    }
    return null;
}
```

Gambar 22. Implementasi dari enkapsulasi

### 7.1.4 Inheritance

*Inheritance* atau pewarisan, di dalam OOP digunakan untuk mengklasifikasikan suatu obyek ke dalam suatu karakteristik dan fungsi yang umum. Maka ini akan membuat bekerja dengan obyek lebih mudah dan menguntungkan, karena hal ini memungkinkan kombinasi karakteristik umum pada obyek induk dengan obyek anak dengan cara mewariskan karakteristik pada obyek induk pada obyek anak. Misalnya suatu kelas Mahasiswa dan Karyawan adalah dua buah obyek yang memiliki karakteristik berbeda, namun kedua obyek tersebut dapat mewarisi obyek induk misalnya Anggota. Karena Mahasiswa dan Karyawan adalah dua obyek yang berbeda tapi keduanya adalah Anggota dari suatu perpustakaan. Dapat dilihat di kode program untuk kelas Mahasiswa Berikut :



```
package entity;  
public class Mahasiswa extends Anggota {  
    private String nim;  
    private int tahunMasuk;  
    public String getNim() {  
        return nim;  
    }  
    public void setNim(String nim) {  
        this.nim = nim;  
    }  
    public int getTahunMasuk() {  
        return tahunMasuk;  
    }  
    public void setTahunMasuk(int tahunMasuk) {  
        this.tahunMasuk = tahunMasuk;  
    }  
}
```

Gambar 23. Implementasi dari *inheritance* untuk kelas mahasiswa

Dan untuk kelas Karyawan

```
package entity;  
public class Karyawan extends Anggota {  
    private String nik;  
    public String getNik() {  
        return nik;  
    }  
    public void setNik(String nik) {  
        this.nik = nik;  
    }  
}
```

Gambar 24. Implementasi dari *inheritance* untuk kelas karyawan

Pada dua kelas diatas, yaitu kelas Mahasiswa dan Karyawan terdapat kode program yang dilingkari yaitu

1. Mahasiswa extends Anggota
2. Karyawan extends Anggota

Dapat dibaca bahwa kelas Mahasiswa adalah turunan dari kelas Anggota dan kelas Karyawan adalah turunan dari kelas Anggota. Apabila anda perhatikan lebih teliti misalnya pada kelas Karyawan maka tampak disana hanya terdapat 1 buah atribut atau field (dilingkari warna merah). Untuk atribut id, nama, tempat lahir dan tanggal lahir cukup dideklarasikan saja di kelas Anggota, sehingga tidak terjadi proses deklarasi variabel yang sama berulang – ulang, berikut ini kode program kelas Anggota.

```
package entity;  
import java.util.Date;  
public class Anggota {  
    private int id;  
    private String nama;  
    private String tmpLahir;  
    private Date tglLahir;  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    /*-----setter dan getter atribut yang lain-----*/  
}
```

Gambar 25. Implementasi dari *inheritance* untuk kelas anggota



Kode program yang dilingkari warna ungu menunjukkan atribut yang dimiliki kelas Anggota seperti id, nama, tmpLahir dan tglLahir. Atribut itu juga yang nantinya akan digunakan oleh kelas Mahasiswa, Karyawan dan PetugasPerpus.

#### **7.1.5 Interface**

Merupakan sebuah keyword untuk mendefinisikan sekumpulan method dan konstanta. Interface bisa digunakan apabila sudah dilakukan implementasi pada sebuah kelas non abstract. Interface dapat digunakan untuk

1. Mewakili suatu tingkah laku yang bisa dimiliki oleh suatu kelas tanpa memaksakan relasi antar kelas tersebut.
2. Menangani event atau action yang terjadi pada suatu kelas
3. Memudahkan penanganan objek dalam parameter
4. Multiple inheritace

#### **7.2 Diagram *Unified Modeling Language* (UML)**

*The Unified Modeling Language* adalah seperangkat aturan dan notasi untuk spesifikasi sistem perangkat lunak, dikelola dan dibuat oleh Object Management Group. notasi ini menyediakan satu set elemen grafis untuk pemodelan sistem.

Beberapa diagram dari UML adalah :

1. *Use Case Diagram*  
Alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram menunjukkan interaksi antara pengguna dan entitas eksternal lainnya dengan sistem yang sedang dikembangkan.
2. *Activity Diagram*  
Menangkap alur dari sebuah sistem, termasuk tindakan utama dan poin keputusan. Diagram ini berguna untuk mendokumentasikan proses bisnis.
3. *Class Diagram*  
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas.
4. *Sequence Diagram*  
Sequence diagram secara khusus menjabarkan sebuah *Use Case*. Diagram ini menunjukkan sejumlah objek dan pesan yang melewati suatu objek.
5. *Component Diagram*  
Komponen diagram digunakan untuk menggambarkan bagaimana komponen suatu sistem yang terhubung bersama di tingkat yang lebih tinggi dari abstraksi dari diagram kelas. Sebuah komponen bisa dimodelkan oleh salah satu atau lebih kelas.
6. *Deployment Diagram*  
Adalah model arsitektur runtime dari sistem dalam pengaturan dunia nyata. Mereka menunjukkan entitas bagaimana perangkat lunak diterapkan ke perangkat fisik.
7. *State Machine Diagram*. Digunakan untuk menggambarkan status transisi dari objek tunggal dalam menanggapi peristiwa.
8. *Interaction Overview Diagrams*  
Merupakan pencangkakan dari *Activity Diagram* dan *Sequence Diagram*. Disini berupa *sequence diagram* yang dipecah menggunakan notasi *activity diagram* untuk menunjukkan aliran kontrol
9. *Communications Diagram*  
Mendesripsikan kumpulan objek yang berinteraksi untuk menjalankan suatu tingkah laku dalam sistem.

Beberapa case tools yang bisa digunakan dalam pembuatan diagram UML antara lain:

1. Visual Paradigm (Komunitas dan Komersil)

2. Violet
3. Tigris UML
4. UMLET
5. Rational Rose
6. JUDE
7. Poseidon UML
8. Visual Studio 2010 (Komersil)

### 7.3 Use Case Diagram (Diagram Use-case)

Industri perangkat lunak telah mempelajari bahwa untuk meraih sukses perencanaan, analisis, desain, konstruksi, dan penyebaran sistem informasi, analisis sistem pertama-tama harus memahami apa saja kebutuhan para stakeholder dan alasan mengapa sistem harus dikembangkan (*user-centered development*/pengembangan berpusatkan pengguna) (Whitten, 2004).

Dengan fokus kepada pengguna sistem, analisis dapat berkonsentrasi untuk mengembangkan bagaimana sistem akan digunakan dan bukan pada bagaimana sistem dibangun. *Use case modelling*/pemodelan use-case adalah sebuah pendekatan yang memfasilitasi pengembangan berpusatkan kegunaan (Whitten, 2004).

Pemodelan use-case awalnya disusun oleh Dr. Ivar Jacobson pada tahun 1986. Diagram use-case merupakan alat komunikasi tingkat tinggi untuk mewakili persyaratan sistem. Diagram menunjukkan interaksi antara pengguna dan entitas eksternal lainnya dengan sistem yang sedang dikembangkan (Whitten, 2004).

Keuntungan atau manfaat dari diagram use-case (Whitten, 2004):

1. Menyediakan tool untuk mencapture persyaratan fungsional
2. Membantu menyusun ulang lingkup sistem menjadi bagian-bagian yang lebih dapat dikelola
3. Menyediakan alat komunikasi dengan para pengguna dan stakeholder yang berhubungan dengan fungsionalitas sistem
4. Memberikan cara bagaimana mengidentifikasi, menetapkan, melacak, mengontrol, dan mengelola kegiatan pengembangan sistem.
5. Menyajikan panduan untuk mengestimasi lingkup, usaha, dan jadwal proyek
6. Menyajikan garis pokok pengujian, khususnya menentukan rencana test dan test case
7. Menyajikan tool untuk melacak persyaratan
8. Menyajikan titik mulai /awal untuk identifikasi objek data atau entitas
9. Menyajikan kerangka kerja untuk mengarahkan proyek pengembangan sistem

Komponen yang ada di diagram use-case (Nugroho, 2009) dan (Fowler, 2004):

1. Use-case

Sebuah use case merupakan unit fungsi yang dapat berinteraksi dengan aktor atau terkait dengan use case lainnya. Penggunaannya diwakili dengan sebuah elips dengan nama use case didalamnya. Nama use case biasanya berupa suatu kata kerja.

Simbol :

Ex : Use Case Presensi



2. Pelaku / aktor

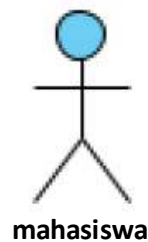
orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

Tipe dari aktor/pelaku adalah :

- a. *Primary business actor* : aktor yang terutama mendapatkan keuntungan dari pelaksanaan use-case (contoh : karyawan dengan menerima gaji untuk periode tertentu)
- b. *Primary system actor* : aktor yang secara langsung berhadapan dengan sistem untuk menginisiasi atau memicu kegiatan atau sistem. (contoh : operator telepon yang memberi bantuan kepada pelanggan, kasir bank yang memproses transaksi)
- c. *External server actor* : aktor yang melayani kebutuhan pengguna use-case. (contoh : biro kredit yang memiliki kuasa atas perubahan kartu kredit).
- d. *External receiver actor* : aktor yang bukan pelaku utama, tapi menerima nilai yang terukur atau teramati (output) dari use-case. (contoh : gudang menerima paket permintaan uantuk menyiapkan pengiriman sesudah seorang pelanggan mememesannya).
- e. *Temporalevent* : kejadian sistem yang dipicu dengan waktu. (contoh : sistem billing untuk perusahaan kartu kredit secara otomatis mencetak tagihan pada hari ke lima dalam bulan itu (tanggal billing); billing PLN; billing PAM, bank merekonsiliasi transaksi tiap hari pada jam 5 sore). Yang menjadi actor/pelaku disini adalah waktu.

Simbol dari aktor digambarkan dengan icon yang mungkin bervariasi namun konsepnya sama yaitu :

- 1) Umumnya untuk orang digambarkan dengan sosok dengan kepala, badan, tangan, dan kaki.
- 2) Umumnya, untuk sistem digambarkan dengan segi empat disertai notasi “<<Actor>>” di atas label nama.

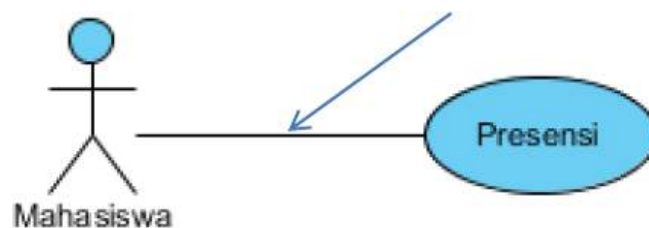


### 3. Relationship / hubungan

Hubungan digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan hubungan berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Macam hubungan di diagram use-case adalah :

#### a. *Association* (gabungan)

*Association* adalah hubungan antara pelaku/actor dengan use case dimana terjadi interaksi di antara keduanya. Diagambarkan sebagai satu garis lurus yang menghubungkan pelaku dan use case.

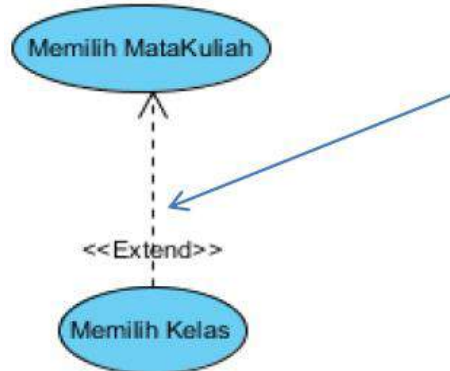


Maknanya : Mahasiswa melakukan presensi

#### b. *Extension*

*Extension use-case* adalah langkah yang diekstraksi dari use-case yang lebih kompleks untuk menyederhanakan dan menyediakan fungsionalitas tambahan kepada usecase induknya pada suatu aliran tertentu. Disini dapat diartikan bahwa use case yang mengextend tidak harus selalu tereksekusi ketika use case induknya dieksekusi. Pada hubungan *extensions*, sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan

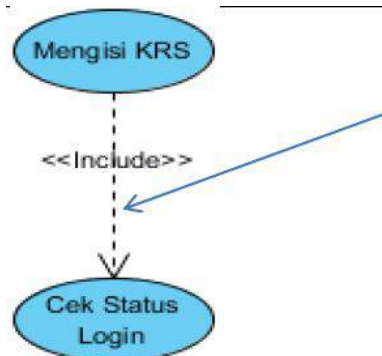
itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.



Maknanya setiap kali use case memilih matakuliah belum tentu use case memilih kelas dieksekusi.

c. *Depens on*

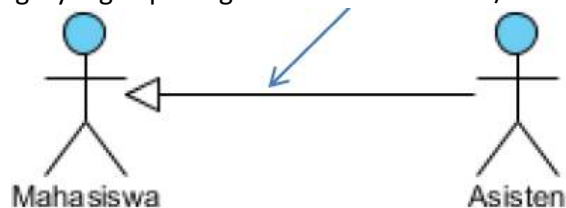
Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.



Maknanya setiap kali ada proses mengisi KRS maka proses pengecekan login pasti selalu dilakukan, jadi ketika Use Case mengisi KRS dieksekusi maka Use Case Cek Status Login akan ikut tereksekusi.

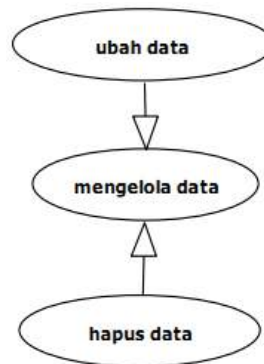
d. *Generalisasi*

Digunakan ketika ada use case yang umum atau actor yang umum, digunakan untuk memberikan dasar fungsi yang dapat digunakan oleh use case/actor yang lebih khusus.



Maknanya mahasiswa merupakan actor yang lebih umum daripada asisten, dimana tidak semua mahasiswa adalah asisten dan semua asisten adalah mahasiswa.

Contoh lainnya :



Maknanya use case mengelola data terdiri dari proses ubah data dan hapus data.

Langkah-langkah membuat use case (Nugroho, 2009) dan (Fowler, 2004):

1. Menemukan aktor

Aktor adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan sesuatu (Kurt Bittner, Ian Spence. 2002). Cara mudah untuk menemukan aktor adalah dengan bertanya hal-hal berikut:

- a. SIAPA yang akan menggunakan sistem?
- b. APAKAH sistem tersebut akan memberikan NILAI bagi aktor?

Namun, yang perlu diingat adalah tidak semua aktor adalah manusia, bisa saja sistem lain yang berinteraksi dengan sistem yang anda buat. Untuk menemukan sistem lain sebagai aktor, hal-hal di bawah ini bisa menjadi pertimbangan.

- a. Jika anda bergantung pada sistem lain untuk melakukan sesuatu, maka sistem lain itu adalah aktor.
- b. Jika sistem lain itu meminta (*request*) informasi dari sistem anda, maka sistem lain itu adalah aktor.

Untuk penamaan, aktor diberi nama sesuai dengan PERAN-nya.

2. Menemukan use case

Sebuah use case harus mendeskripsikan sebuah pekerjaan dimana pekerjaan tersebut akan memberikan NILAI yang bermanfaat bagi aktor (Kurt Bittner, Ian Spence. 2002).

Untuk menemukan use case, mulailah dari sudut pandang aktor, misalnya dengan bertanya :

- a. Informasi apa sajakah yang akan didapatkan aktor dari sistem?
- b. Apakah ada kejadian dari sistem yang perlu diberitahukan ke aktor?

Sedangkan dari sudut pandang sistem, misalnya dengan pertanyaan sebagai berikut :

- a. Apakah ada informasi yang perlu disimpan atau diambil dari sistem?
- b. Apakah ada informasi yang harus dimasukkan oleh aktor?

3. Membuat skenario per-use case dan semua alternatifnya.

4. Definisikan asosiasi antara setiap actor dan setiap use case.

5. Evaluasi setiap actor dan setiap use case untuk mendapatkan kemungkinan refinement (penjelasan lebih detail tentang hubungan setiap aktor dan use casenya).

6. Evaluasi setiap use case untuk dependensi <<include>>.

7. Evaluasi setiap use case untuk dependensi <<extend>>.

8. Evaluasi setiap actor dan setiap use case untuk generalisasi.

9. Langkah terakhir adalah mulai menggambar use casenya. Berikut adalah gambar use case untuk **sistem informasi perpustakaan AMIKOM**.

Contoh kasus :

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begi tu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para pengunjung perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.

Solusi :

1. Menemukan aktor

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia, CD. Sementara itu dosen diijinkan meminjam semua koleksi di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. Para **pengunjung (mahasiswa, dosen, dan karyawan)** perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu **petugas perpustakaan** juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.

Dari ilustrasi di kasus tersebut dapat disimpulkan aktor yang terlibat dalam sistem yang akan dibangun adalah :

- a. Mahasiswa
- b. Dosen
- c. Karyawan
- d. Petugas perpustakaan

2. Menemukan use case

Perpustakaan Universitas AMIKOM Yogyakarta menyewakan buku dan CD kepada mahasiswa, dosen, dan karyawan. Dimana seorang **mahasiswa hanya diijinkan meminjam buku berbahasa Indonesia**, CD. Sementara itu **dosen diijinkan meminjam semua koleksi** di perpustakaan termasuk buku asing, begitu juga karyawan. Seorang mahasiswa hanya diijinkan meminjam maksimal 3 buah koleksi, berbeda dengan dosen dan karyawan yang bisa meminjam tanpa batas. **Para pengunjung (mahasiswa, dosen, dan karyawan) perpustakaan dapat melakukan pencarian koleksi berdasarkan judul, pengarang atau penerbit**, melihat daftar koleksi di komputer pencarian. Prosedur peminjaman adalah dengan membawa buku yang hendak dipinjam dan menunjukkan KTM untuk mahasiswa, NIK untuk dosen dan karyawan kepada petugas perpustakaan. **Mahasiswa diperbolehkan meminjam selama satu minggu, selebihnya didenda. Petugas perpustakaan bertugas melakukan pendataan buku, mencatat peminjaman dan pengembalian, menghitung denda, serta membantu mencarikan koleksi** melalui mesin pencari. Para pengunjung perpustakaan dapat meminjam buku, mencari koleksi di mesin pencari dan membayar denda. Selain itu **petugas perpustakaan juga dapat mencetak laporan seperti Laporan Daftar Koleksi, Laporan Transaksi Harian, Bulanan, Laporan Daftar Anggota, Laporan Pembayaran Denda.**

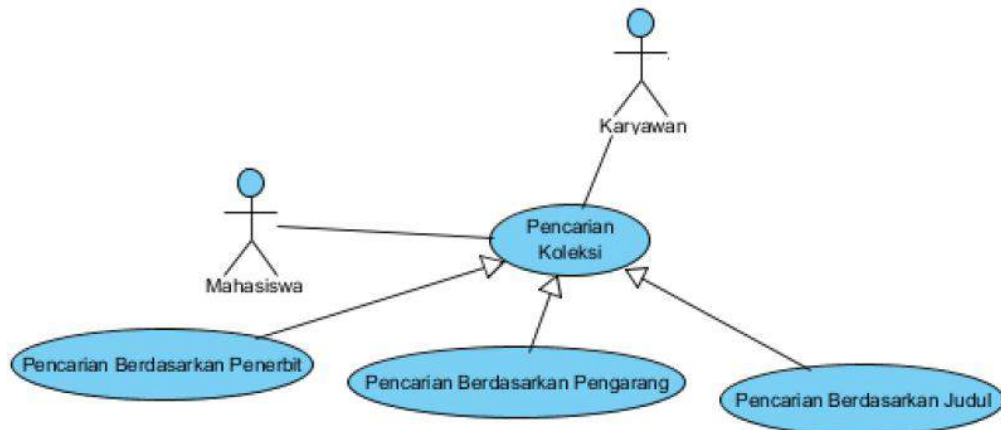
Dari ilustrasi di kasus tersebut use-case yang dibuat dalam sistem yang akan dibangun adalah :

- a. Pencarian koleksi
- b. pencarian berdasarkan penerbit
- c. pencarian berdasarkan pengarang
- d. pencarian berdasarkan judul
- e. meminjam koleksi

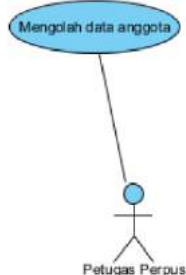
- f. meminjam koleksi berjenis non referensi
  - g. meminjam koleksi berjenis referensi
  - h. mengolah data laporan : laporan denda, laporan transaksi harian, laporan koleksi, laporan daftar anggota
  - i. login
  - j. mengolah data koleksi
  - k. mencatat transaksi peminjaman
  - l. mengecek apakah masih ada buku yang dipinjam
  - m. mencatat transaksi pengembalian
  - n. mencatat denda pengembalian
  - o. mengecek keterlambatan pengembalian
3. Membuat skenario per-use case dan semua alternatifnya.
  4. Definisikan asosiasi antara setiap actor dan setiap use case.
  5. Evaluasi setiap actor dan setiap use case untuk mendapatkan kemungkinan *refinement* (penjelasan lebih detail tentang hubungan setiap aktor dan use casenya).
  6. Evaluasi setiap use case untuk dependensi <<include>>.
  7. Evaluasi setiap use case untuk dependensi <<extend>>.
  8. Evaluasi setiap actor dan setiap use case untuk generalisasi.

Untuk poin 3 sampai dengan 4, penggambaran penyelesaian kasus dapat dijabarkan sebagai berikut :

1. mahasiswa dan karyawan bisa melakukan pencarian koleksi berdasarkan : penerbit, pengarang, judul

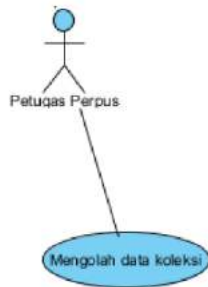


2. petugas perpustakaan bisa melakukan olah data anggota

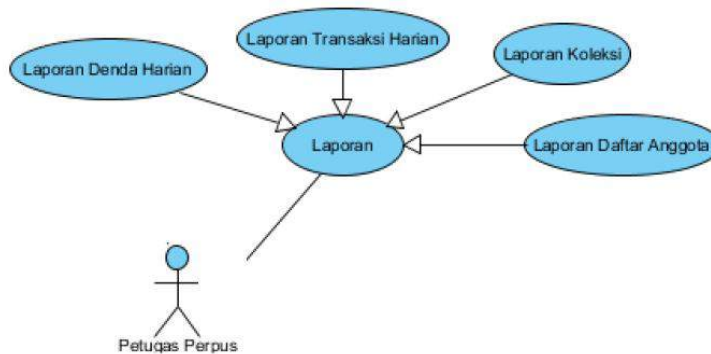




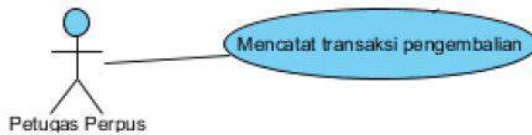
3. petugas perpustakaan bisa melakukan olah data koleksi



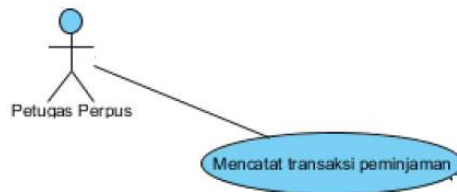
4. petugas perpustakaan bisa melakukan cetak laporan denda harian, transaksi harian, koleksi, dan laporan daftar anggota



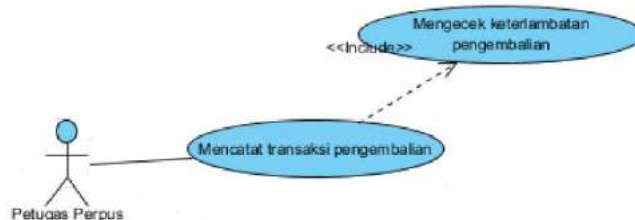
5. petugas perpustakaan bisa melakukan pencatatan transaksi pengembalian



6. petugas perpustakaan bisa mencatat transaksi peminjaman



7. proses pencatatan transaksi pengembalian harus melakukan pengecekan keterlambatan pengembalian terlebih dahulu

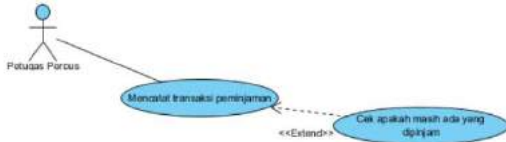




8. ketika mencatat transaksi peminjaman, bisa juga melakukan pencatatan denda pengembalian apabila mahasiswa terlambat mengembalikan

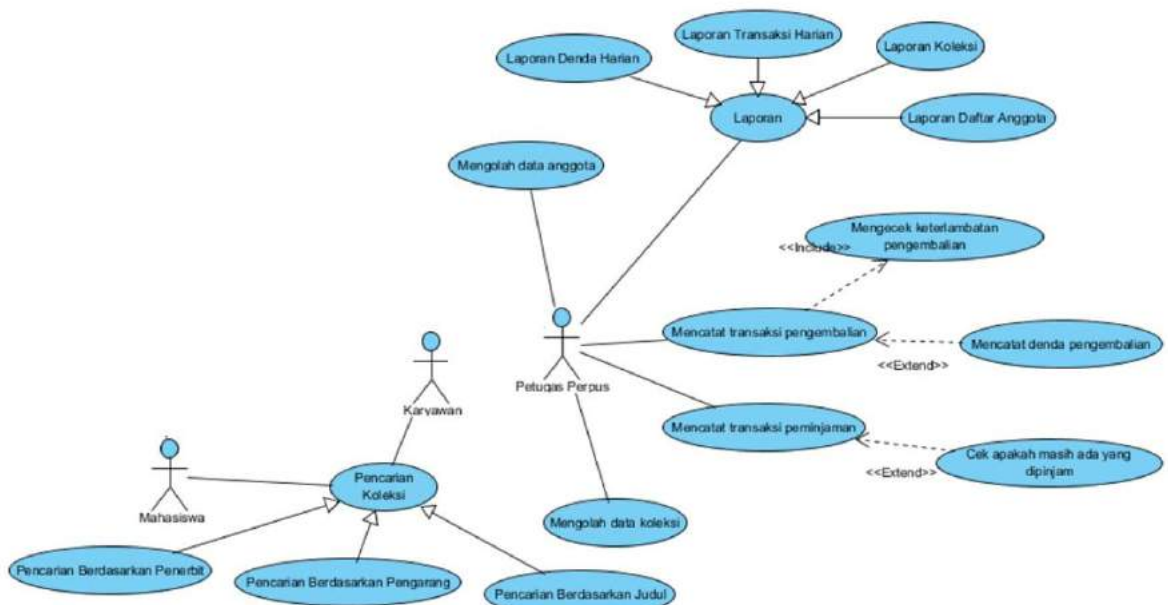


9. ketika mencatat transaksi peminjaman, petugas perpustakaan juga bisa melakukan pengecekan apakah masih ada buku yang dipinjam dan belum dikembalikan



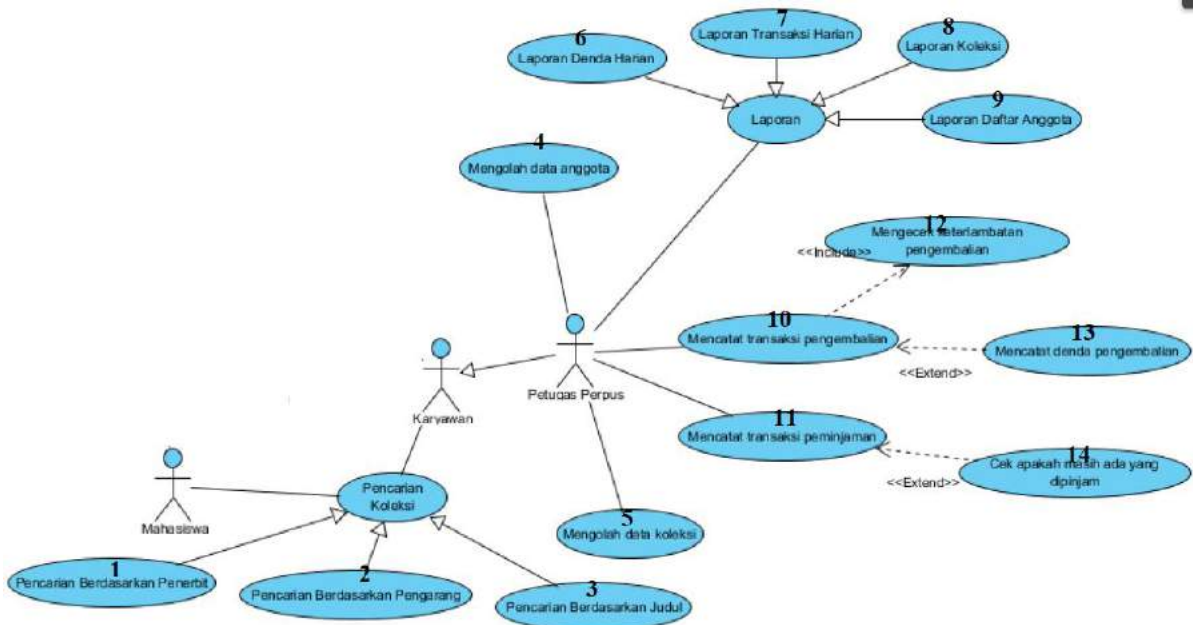
9. Mulai menggambar use casenya secara keseluruhan

### Use Case Diagram Perpustakaan



Gambar 26. Use case diagram untuk sistem informasi perpustakaan AMIKOM

10. Yang terakhir adalah membuat deskripsi tiap use-case



**Gambar 27.** Penomoran use case diagram untuk sistem informasi perpustakaan AMIKOM

Deskripsi use case dari gambar use case di atas bisa dilihat di tabel 10 Sampai dengan tabel 23.

**Tabel 10.** Deskripsi use case pencarian berdasarkan penerbit

No. Use-Case	:	1
Nama Use Case	:	Pencarian berdasarkan penerbit
Actor	:	Mahasiswa, karyawan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencarian buku berdasarkan nama penerbit
Pre Condition	:	Form pencarian telah terbuka
Post Condition	:	Menampilkan data buku berdasarkan penerbit yang dicari
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memasukkan nama penerbit yang akan dicari		
2. Klik tombol CARI		
		3. Sistem akan melakukan pengecekan ke table buku dan penerbit
		4. Jika nama penerbit tidak ditemukan, maka system akan menampilkan MESSAGE "MAAF BUKU YANG DICARI TIDAK DITEMUKAN"
		5. Jika data penerbit ditemukan, maka system akan menampilkan buku berdasarkan keyword penerbit

**Tabel 11.** Deskripsi use case pencarian berdasarkan pengarang

No. Use-Case	:	2
Nama Use Case	:	Pencarian berdasarkan pengarang
Actor	:	Mahasiswa, karyawan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencarian buku berdasarkan nama pengarang
Pre Condition	:	Form pencarian telah terbuka
Post Condition	:	Menampilkan data buku berdasarkan pengarang yang dicari
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memasukkan nama pengarang yang akan dicari		
2. Klik tombol CARI		
		3. Sistem akan melakukan pengecekan ke table buku dan pengarang
		4. Jika nama pengarang tidak ditemukan, maka system akan menampilkan MESSAGE "MAAF BUKU YANG DICARI TIDAK DITEMUKAN"
		5. Jika data pengarang ditemukan, maka system akan menampilkan buku berdasarkan keyword pengarang

**Tabel 12.** Deskripsi use case pencarian berdasarkan judul

No. Use-Case	:	3
Nama Use Case	:	Pencarian berdasarkan judul
Actor	:	Mahasiswa, karyawan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencarian buku berdasarkan judul
Pre Condition	:	Form pencarian telah terbuka
Post Condition	:	Menampilkan data buku berdasarkan judul yang dicari
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memasukkan judul yang akan dicari		
2. Klik tombol CARI		
		3. Sistem akan melakukan pengecekan ke table buku dan judul
		4. Jika judul tidak ditemukan, maka system akan menampilkan MESSAGE "MAAF BUKU YANG DICARI TIDAK DITEMUKAN"
		5. Jika data judul ditemukan, maka system akan menampilkan buku berdasarkan keyword judul

**Tabel 13.** Deskripsi use case mengolah data anggota

No. Use-Case	:	4
Nama Use Case	:	Mengolah data anggota
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pendaftaran dan pengolahan data anggota
Pre Condition	:	Form pengolahan data anggota telah terbuka
Post Condition	:	Data anggota tersimpan di table anggota

<b>Alur Berjalannya Use-Case Input Data Anggota</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor mengisikan data anggota yang mendaftar	
2. Klik tombol SIMPAN	
	3. Sistem akan melakukan pengecekan data yang diisikan apakah telah sesuai atau belum
	4. Jika ada yang belum diisi, maka system akan menampilkan MESSAGE "MAAF SILAHKAN LENGKAPI DATA"
	5. Jika data yang diisikan telah sesuai, maka data anggota akan tersimpan di table anggota
Alternatif : - update data anggota	
<b>Alur Berjalannya Use-Case Update Data Anggota</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memilih data anggota yang akan diubah dan melakukan perubahan pada data	
2. Klik tombol UPDATE	
	3. Sistem akan melakukan pengecekan data yang diubah apakah telah sesuai atau belum
	4. Jika ada yang belum diisi, maka system akan menampilkan MESSAGE "MAAF SILAHKAN LENGKAPI DATA"
	5. Jika data yang diisikan telah sesuai, maka data anggota akan tersimpan di table anggota
Alternatif : - delete data anggota	
<b>Alur Berjalannya Use-Case Hapus Data Anggota</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memilih data anggota yang akan dihapus	
2. Klik tombol HAPUS	
	3. Sistem akan menampilkan message "YAKIN DATA INI AKAN DIHAPUS ?"
4. Aktor klik button YA	
	5. Data yang dihapus akan hilang dari table anggota

**Tabel 14.** Deskripsi use case mengolah data koleksi

No. Use-Case	:	5
Nama Use Case	:	Mengolah data koleksi
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari prose pengolahan (input, update, delete data anggota)
Pre Condition	:	Form pengolahan data koleksi telah terbuka
Post Condition	:	Data koleksi tersimpan di table koleksi
<b>Alur Berjalannya Use-Case Input Data Koleksi</b>		
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>	
1. Aktor mengisikan data koleksi buku yang masuk		
2. Klik tombol SIMPAN		
	3. Sistem akan melakukan pengecekan data yang diisikan apakah telah lengkap atau belum	

	4. Jika ada yang belum diisi, maka system akan menampilkan MESSAGE "MAAF SILAHKAN LENGKAPI DATA"
	5. Jika data yang diisikan telah sesuai, maka data akan tersimpan di table koleksi
Alternatif : - update data koleksi	
<b>Alur Berjalannya Use-Case Update Data Koleksi</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memilih data koleksi yang akan diubah dan melakukan perubahan pada data	
2. Klik tombol UPDATE	
	3. Sistem akan melakukan pengecekan data yang diubah apakah telah sesuai atau belum
	4. Jika ada yang belum diisi, maka system akan menampilkan MESSAGE "MAAF SILAHKAN LENGKAPI DATA"
	5. Jika data yang diisikan telah sesuai, maka data akan tersimpan di table koleksi
Alternatif : - delete data koleksi	
<b>Alur Berjalannya Use-Case Hapus Data Koleksi</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor memilih data koleksi yang akan dihapus	
2. Klik tombol HAPUS	
	3. Sistem akan menampilkan message "YAKIN DATA INI AKAN DIHAPUS ?"
4. Aktor klik button YA	
	5. Data yang dihapus akan hilang dari table koleksi

**Tabel 15.** Deskripsi use case laporan denda harian

No. Use-Case	:	6
Nama Use Case	:	Laporan denda harian
Actor	:	Petugas Perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari proses pencetakan laporan denda harian
Pre Condition	:	Petugas perpustakaan telah memilih tanggal yang akan dicetak laporan denda hariannya
Post Condition	:	Laporan denda harian tampil
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>	
1. Aktor memilih tanggal		
2. Klik tombol OK		
	3. Sistem akan menjalankan query untuk mengambil data di tabel denda sesuai dengan tanggal yang dimasukkan	
	4. Sistem menampilkan laporan denda harian sesuai tanggal yang dimasukkan	

**Tabel 16.** Deskripsi use case laporan transaksi harian

No. Use-Case	:	7
Nama Use Case	:	Laporan transaksi harian
Actor	:	Petugas Perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari proses pencetakan laporan transaksi peminjaman dan pengembalian
Pre Condition	:	Petugas perpustakaan telah memilih tanggal yang akan dicetak laporan transaksi hariannya
Post Condition	:	Laporan transaksi harian tampil
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memilih tanggal		
2. Klik tombol OK		
		3. Sistem akan menjalankan query untuk mengambil data di tabel anggota, TransaksiPengembalian, TransaksiPeminjaman sesuai dengan tanggal yang dimasukkan
		4. Sistem menampilkan laporan transaksi harian sesuai tanggal yang dimasukkan

**Tabel 17.** Deskripsi use case laporan koleksi

No. Use-Case	:	8
Nama Use Case	:	Laporan koleksi
Actor	:	Petugas Perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari proses pencetakan laporan koleksi
Pre Condition	:	Petugas perpustakaan telah memilih jenis koleksi yang akan dicetak laporannya
Post Condition	:	Laporan koleksi tampil
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memilih jenis koleksi		
2. Klik tombol OK		
		3. Sistem akan menjalankan query untuk mengambil data di tabel koleksi dan tabel buku sesuai dengan jenis koleksi yang dimasukkan
		4. Sistem menampilkan laporan koleksi sesuai jenis koleksi yang dimasukkan

**Tabel 18.** Deskripsi use case laporan daftar anggota

No. Use-Case	:	9
Nama Use Case	:	Laporan daftar anggota
Actor	:	Petugas Perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari proses pencetakan laporan anggota per periode
Pre Condition	:	Petugas perpustakaan telah memilih rentang tanggal yang akan dicetak laporannya
Post Condition	:	Laporan daftar anggota tampil
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor memilih rentang tanggal		

2. Klik tombol OK	
	3. Sistem akan menjalankan query untuk mengambil data di tabel anggota sesuai dengan rentang tanggal yang dimasukkan
	4. Sistem menampilkan laporan daftar anggota sesuai rentang tanggal yang dimasukkan

**Tabel 19.** Deskripsi use case mencatat transaksi pengembalian

No. Use-Case	:	10
Nama Use Case	:	Mencatat transaksi pengembalian
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencatatan transaksi pengembalian
Pre Condition	:	1. Form pengembalian buku telah terbuka 2. Sistem telah mengecek keterlambatan pengembalian
Post Condition	:	Data transaksi pengembalian tersimpan di tabel
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor mengisi kode koleksi		
		2. sistem akan mengambil data koleksi di tabel TransaksiPeminjaman
		3. Sistem akan menghitung jumlah keterlambatan dan akan memunculkannya
		4. Sistem akan menghitung denda keterlambatan dan menampilkannya
5. Klik tombol SIMPAN		
		9. sistem akan mengecek data pengembalian. Jika valid maka sistem akan menyimpan data tersebut ke tabel transaksiPengembalian
		10. Jika tidak valid, sistem akan menampilkan message
Alternatif : - mengecek keterlambatan pengembalian buku		

**Tabel 20.** Deskripsi use case mencatat transaksi peminjaman

No. Use-Case	:	11
Nama Use Case	:	Mencatat transaksi peminjaman
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencatatan transaksi peminjaman
Pre Condition	:	Form peminjaman buku telah terbuka
Post Condition	:	Data transaksi peminjaman tersimpan di tabel
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor mengisi nomor anggota (nim)		
2. Klik tombol PROSES		
		3. Sistem akan melakukan pengecekan data nomor anggota
		4. sistem akan memunculkan nama anggota
		5. sistem mengaktifkan form input peminjaman

6. Aktor mengisikan data koleksi yang akan dipinjam	
	7. Sistem akan melakukan pengecekan data koleksi yang dipinjam. Jika lebih dari 3, maka sistem akan mengeluarkan message yang isinya peminjaman maksimal berjumlah 3.
	8. Jika peminjaman koleksi memenuhi maka sistem akan memunculkan data koleksi
9. Klik tombol SIMPAN	
	9. sistem akan mengecek data peminjaman. Jika valid maka sistem akan menyimpan data tersebut ke tabel transaksiPeminjaman
	10. Jika tidak valid, sistem akan menampilkan message
Alternatif : - pengecekan buku yang masih dipinjam	

**Tabel 21.** Deskripsi use case mengecek keterlambatan pengembalian

No. Use-Case	:	12
Nama Use Case	:	Mengecek keterlambatan pengembalian
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari system melakukan pengecekan keterlambatan peminjaman buku
Pre Condition	:	Sistem telah memasukkan data pengembalian
Post Condition	:	Data keterlambatan transaksi peminjaman tersimpan di tabel pengembalian
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
		1. Sistem mengecek tanggal pinjam dengan tanggal kembali
		2. Sistem melakukan pengurangan tanggal pinjam dengan tanggal kembali untuk mendapatkan jumlah hari yang terlambat
		3. sistem akan menyimpan iddenda ke tabel pengembalian
		4. sistem menampilkan daftar pengembalian buku dan denda
Alternatif : - mencatat denda pengembalian		

**Tabel 22.** Deskripsi use case mencatat denda pengembalian

No. Use-Case	:	13
Nama Use Case	:	Mencatat denda pengembalian
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk mencatat denda pengembalian buku
Pre Condition	:	Data pengembalian telah diinputkan
Post Condition	:	Data denda tersimpan di tabel denda
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
		1. Sistem mengecek tanggal pinjam dengan tanggal kembali
		2. Sistem melakukan pengurangan tanggal pinjam dengan tanggal kembali untuk mendapatkan jumlah hari yang terlambat



	3. sistem akan menyimpan iddenda dan nominalnya ke tabel denda
	4. sistem menginputkan iddenda ke tabel transaksiPengembalian dan menampilkan nominalnya ke form

**Tabel 23.** Deskripsi use case mengecek buku yang masih dipinjam

No. Use-Case	:	14
Nama Use Case	:	Mengecek buku yang masih dipinjam
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari system melakukan pengecekan buku yang masih dipinjam
Pre Condition	:	Sistem telah memvalidasi nomor anggota
Post Condition	:	Data peminjaman tersimpan di tabel
<b>Alur Berjalannya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
		1. Sistem mengecek peminjaman buku yang belum dikembalikan berdasarkan nomor anggota
		2. Sistem menampilkan buku yang masih dipinjam
Alternatif : - mencatat denda pengembalian		





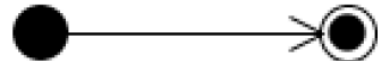
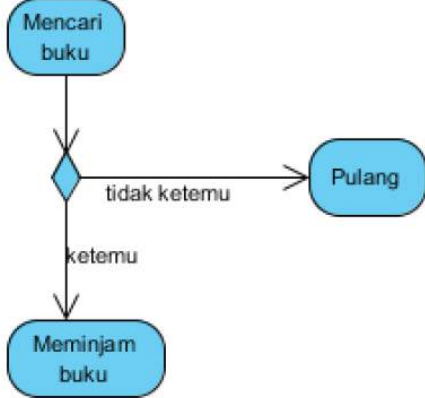
#### 7.4 Activity Diagram

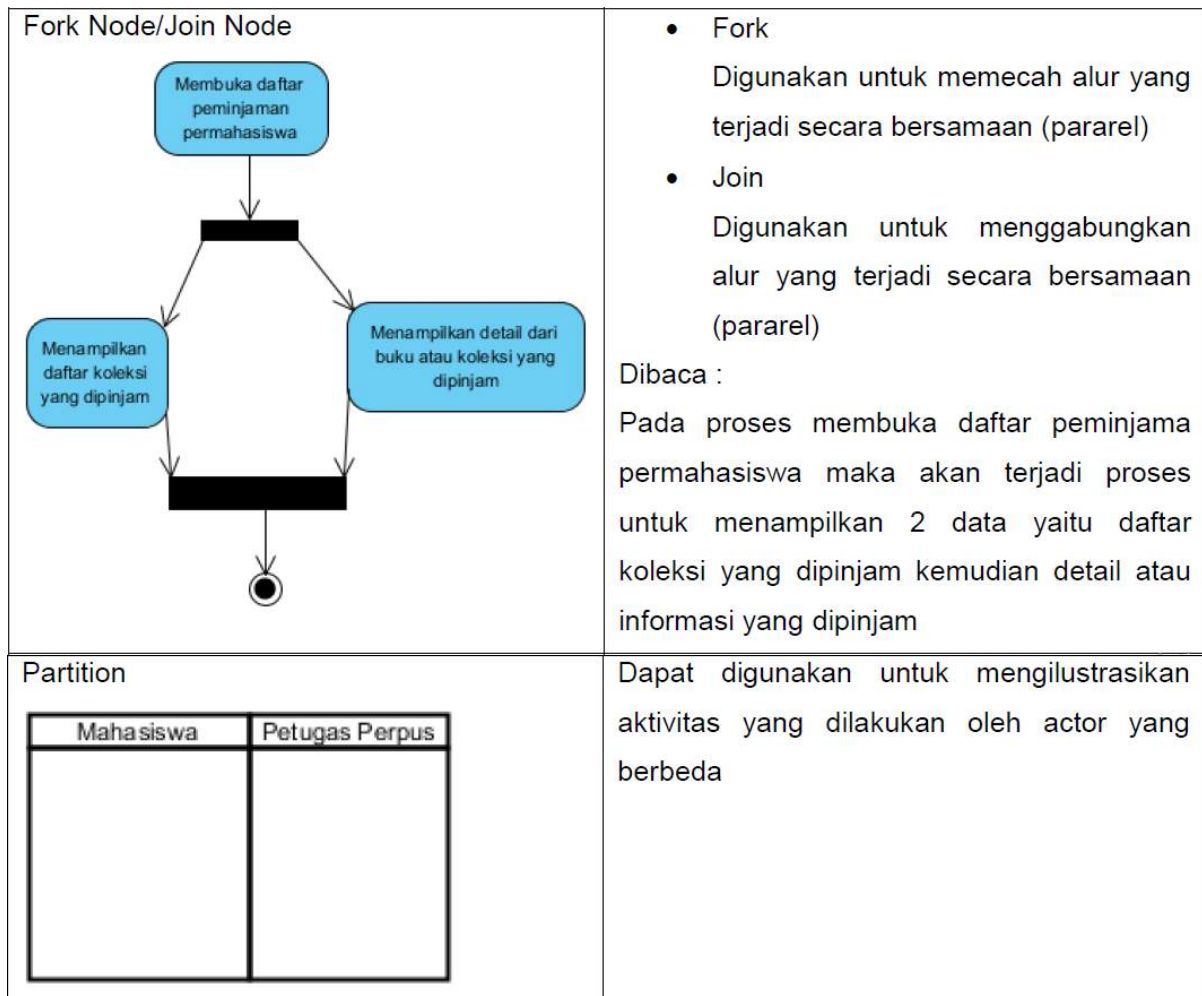
Diagram aktivitas atau *activity diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem [14].

Fungsi *activity diagram* adalah [14]:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya

Simbol dari activity diagram dapat dilihat pada gambar di bawah ini:

<p>Action</p> 	<p>Menggambarkan 1 langkah dalam alur system.</p>
<p>Start Node/Initial Node</p> 	<p>Digunakan untuk menggambarkan dimana aliran berawal</p>
<p>Activity Final Node</p> 	<p>Akhir dari banyak aliran dalam diagram aktivitas</p>
<p>Flow Final Node</p> 	<p>Menunjukkan akhir dari aliran tunggal dalam diagram aktivitas</p>
<p>Control Flow</p> 	<p>Menunjukkan aliran kendali dari satu aksi ke aksi yang berikutnya lain.</p>
<p>Decision Node</p>	<p>Digunakan untuk merepresentasikan keputusan dalam alur kendali. Ini juga bisa digunakan untuk menggabungkan alur.</p>
	<p>Sebuah keputusan akan memiliki kondisi yang harus ada untuk menentukan jalan atau alur yang akan diambil</p> <p>Dibaca :</p> <p>Pada sebuah kegiatan mencari buku ada proses penentuan keputusan, apabila buku tidak ketemu pulang, ketemu maka meminjam</p>



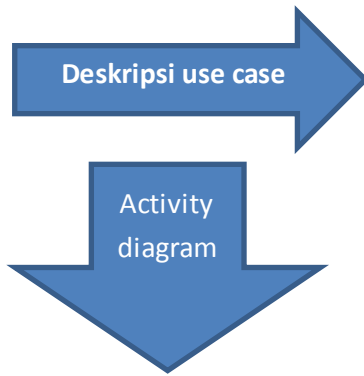
**Gambar 28.** Simbol Activity Diagram

Langkah membuat *activity diagram* adalah (Nugroho, 2009) dan (Fowler, 2004):

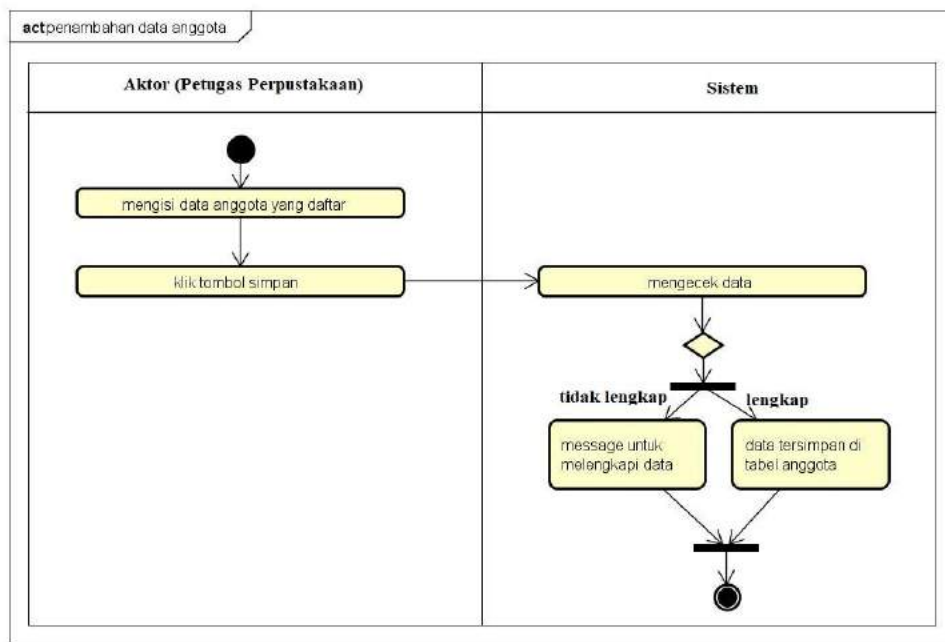
- a. Buat simbol status awal ketika mengawali diagram
- b. Gambarkan aksi pertama dan seterusnya sesuai aliran kegiatan sistem. Gunakan sebuah fork ketika berbagai aktivitas terjadi secara bersamaan. Setelah penggabungan seluruh kegiatan paralel, harus digabungkan dengan simbol join.
- c. Cabang keputusan digunakan untuk menunjukkan suatu kegiatan yang memenuhi kondisi tertentu. Seluruh pancabangan diakhiri tanda penggabungan (menggunakan tanda decision) sebagai akhir perilaku tersebut.
- d. Akhiri diagram dengan simbol status akhir.
- e. Jumlah dan alur dari *activity diagram* harus sama dengan jumlah use-case dan deskripsi use-case

Modul ini hanya mencontohkan 3 activity saja yaitu : penambahan data anggota, peminjaman buku dan pengembalian buku. Untuk lainnya silahkan anda buat :

1. Penambahan data anggota  
Pembuatan *activity diagram* harus menyesuaikan dengan alur di deskripsi use case input data anggota.



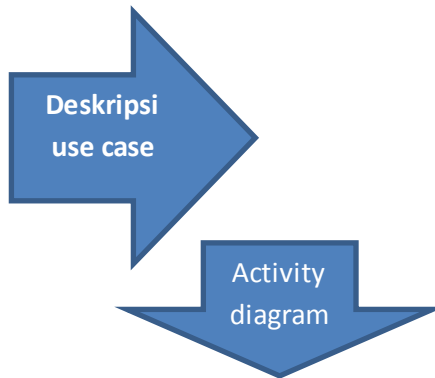
No. Use-Case	: 4
Nama Use Case	: Mengolah data anggota
Aktor	: Petugas perpustakaan
Deskripsi	: Use case ini digunakan untuk menjelaskan alur dari pendaftaran dan pengolahan data anggota
Pre Condition	: Form pengolahan data anggota telah terbuka
Post Condition	: Data anggota tersimpan di table anggota
<b>Alur Berjalannya Use-Case Input Data Anggota</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
1. Aktor mengisi data anggota yang mendaftar	
2. Klik tombol SIMPAN	
	3. Sistem akan melakukan pengecekan data yang diisikan apakah telah sesuai atau belum
	4. Jika ada yang belum diisi, maka system akan menampilkan MESSAGE "MAAF SILAHKAN LENGKAPI DATA"
	5. Jika data yang diisikan telah sesuai, maka data anggota akan tersimpan di table anggota



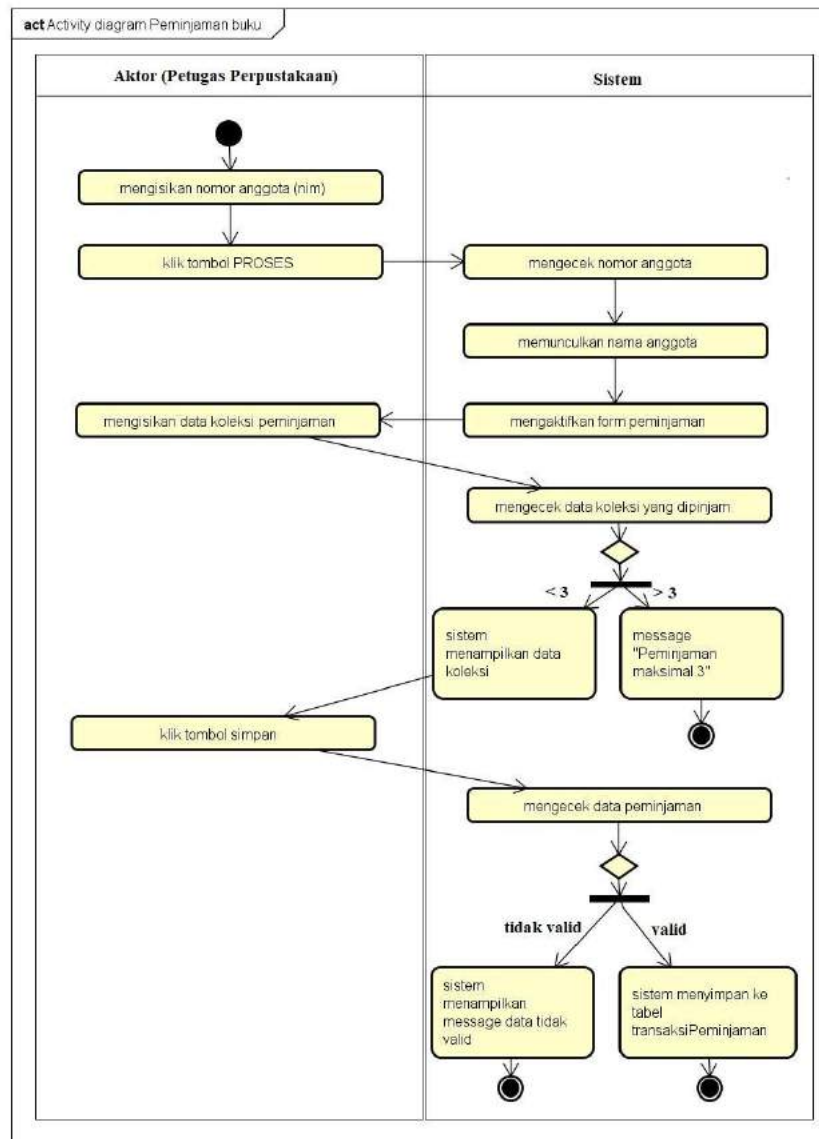
powered by Astah

**Gambar 29.** Activity diagram mengolah data anggota

2. Peminjaman buku  
Pembuatan activity diagram harus menyesuaikan dengan alur di deskripsi use case mencatat transaksi peminjaman.



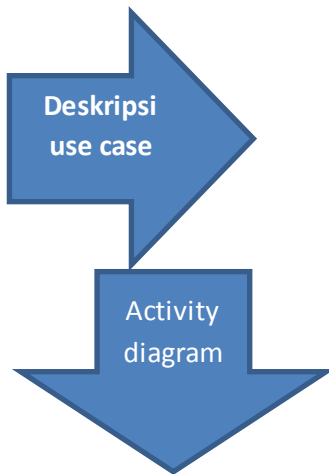
No. Use-Case	: 11
Nama Use Case	: Mencatat transaksi peminjaman
Actor	: Petugas perpustakaan
Deskripsi	: Use case ini digunakan untuk menjelaskan alur dari pencatatan transaksi peminjaman
Pre Condition	: Form peminjaman buku telah terbuka
Post Condition	: Data transaksi peminjaman tersimpan di tabel
Alur Berjalananya Use-Case	
Aksi Aktor	Reaksi Sistem
1. Aktor mengisikan nomor anggota (nim)	
2. Klik tombol PROSES	
	3. Sistem akan melakukan pengecekan data nomor anggota
	4. sistem akan memunculkan nama anggota
	5. sistem mengaktifkan form input peminjaman
6. Aktor mengisikan data koleksi yang akan dipinjam	
	7. Sistem akan melakukan pengecekan data koleksi yang dipinjam. Jika lebih dari 3, maka sistem akan mengeluarkan message yang isinya peminjaman maksimal berjumlah 3.
	8. Jika peminjaman koleksi memenuhi maka sistem akan memunculkan data koleksi
9. Klik tombol SIMPAN	
	9. sistem akan mengecek data peminjaman. Jika valid maka sistem akan menyimpan data tersebut ke tabel transaksiPeminjaman
	10. Jika tidak valid, sistem akan menampilkan message



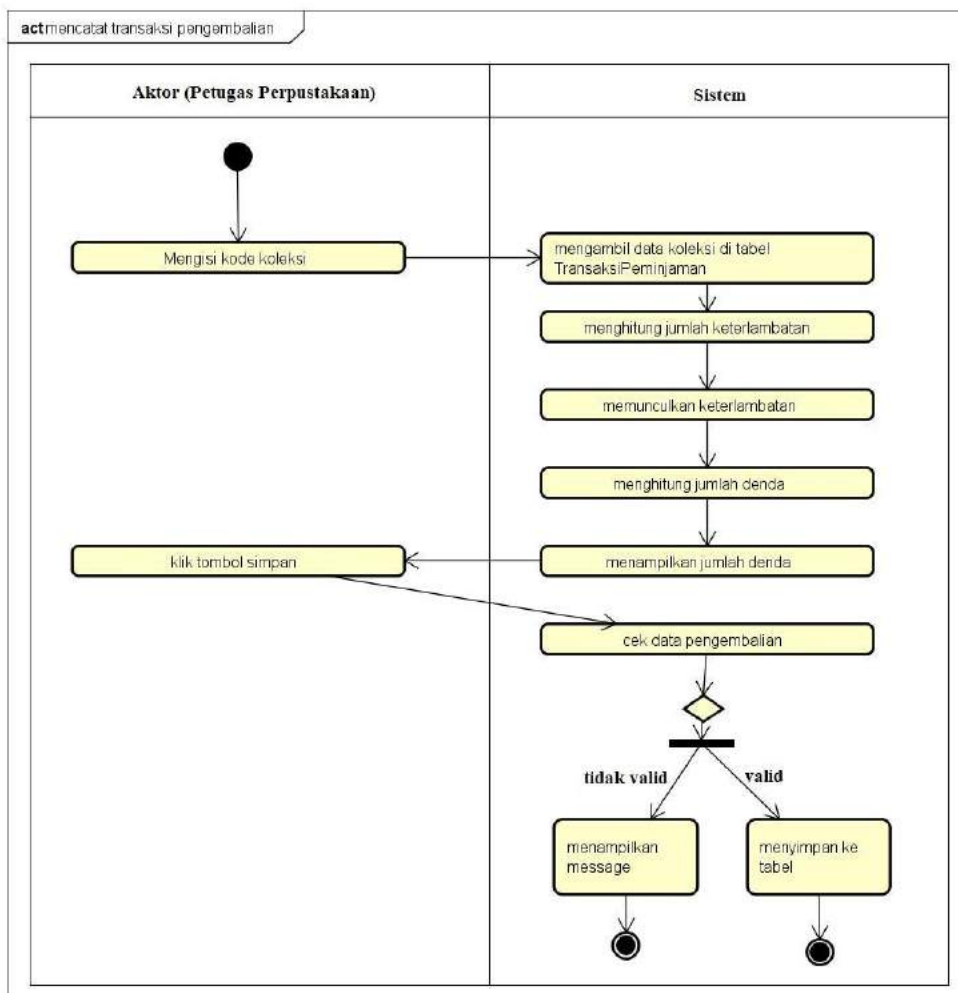
**Gambar 30.** Activity diagram mencatat transaksi peminjaman

3. Pengembalian buku

Pembuatan activity diagram harus menyesuaikan dengan alur di deskripsi use case mencatat pengembalian.



No. Use-Case	:	10
Nama Use Case	:	Mencatat transaksi pengembalian
Actor	:	Petugas perpustakaan
Deskripsi	:	Use case ini digunakan untuk menjelaskan alur dari pencatatan transaksi pengembalian
Pre Condition	:	1. Form pengembalian buku telah terbuka 2. Sistem telah mengecek keterlambatan pengembalian
Post Condition	:	Data transaksi pengembalian tersimpan di tabel
<b>Alur Berjalananya Use-Case</b>		
<b>Aksi Aktor</b>		<b>Reaksi Sistem</b>
1. Aktor mengisikan kode koleksi		2. sistem akan mengambil data koleksi di tabel TransaksiPeminjaman
		3. Sistem akan menghitung jumlah keterlambatan dan akan memunculkannya
		4. Sistem akan menghitung denda keterlambatan dan menampilkan
5. Klik tombol SIMPAN		9. sistem akan mengecek data pengembalian. Jika valid maka sistem akan menyimpan data tersebut ke tabel transaksiPengembalian
		10. Jika tidak valid, sistem akan menampilkan message



powered by Astah

**Gambar 31.** Activity diagram mencatat transaksi pengembalian

### 7.5 Class Diagram

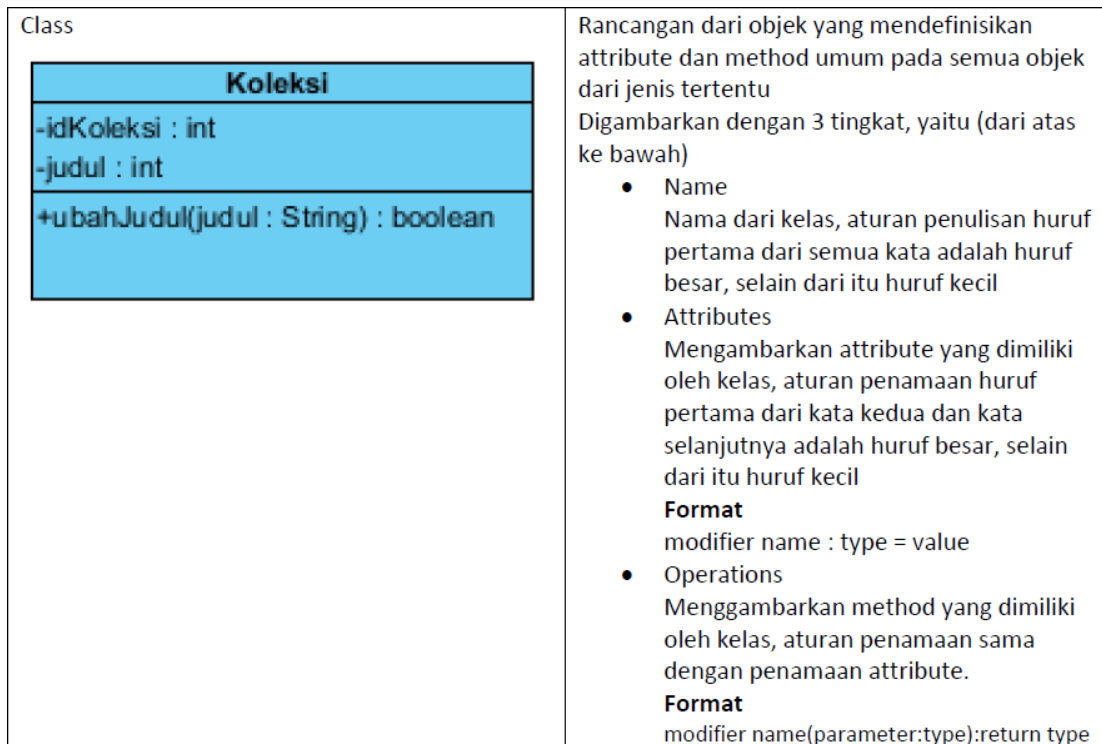
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Fowler, 2004).

- atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
- operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

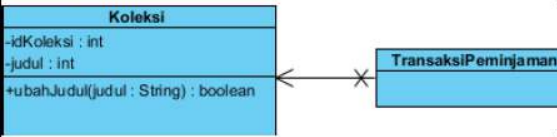
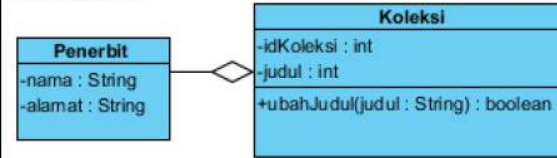
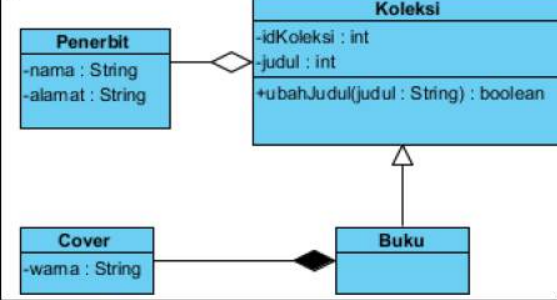
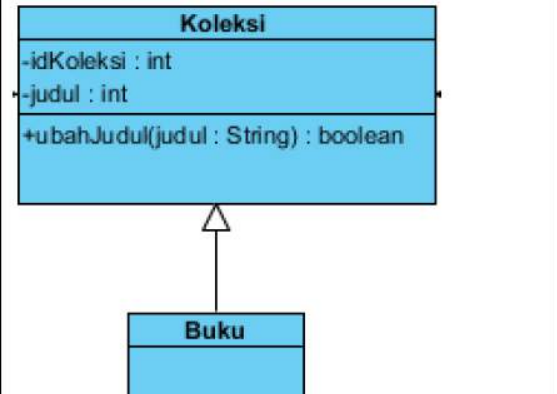
Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut (Fowler, 2004).

Diagram kelas menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Simbol dari class diagram dapat dilihat pada gambar di bawah ini (Fowler, 2004):





<p>Association</p> 	<p>Relasi yang kuat, relasi ini mengindikasikan bahwa suatu kelas mereferensikan kelas yang lain, menggambarkan interaksi yang mungkin terjadi antara 1 kelas dengan kelas yang lain selama kelas tersebut tidak saling memiliki atau bukan bagian dari.</p> <ul style="list-style-type: none"> <li>• Directional Association (1 arah) Menggambarkan bahwa pesan terjadi hanya dari satu kelas sedangkan kelas yang lain pasif</li> <li>• Bidirectional Association (2 arah) Menggambarkan bahwa pesan dari kedua kelas yang saling berelasi.</li> </ul>
<p>Aggregation</p> 	<p>Lebih spesifik dari asosiasi, mengindikasikan bahwa suatu kelas merupakan bagian dari kelas yang lain namun bersifat tidak wajib. Relasi ini juga menyatakan bahwa suatu kelas yang menjadi bagian dari kelas yang lain tidak akan dihapus meskipun kelas yang memilikinya dihapus.</p>
	<p>bagian dari kelas yang lain dan bersifat wajib. Relasi ini juga mengindikasikan bahwa suatu kelas yang menjadi bagian kelas yang lain akan terhapus ketika kelas yang memilikinya dihapus.</p>
<p>Generalization</p> 	<p>Juga dikenal sebagai inheritance. Mengindikasikan bahwa sebuah kelas adalah child class atau sub class lebih spesifik terhadap parent class / super kelasnya. Semua sub class akan memiliki apa yang dimiliki super class tapi super class tidak memiliki apa yang hanya dimiliki oleh sub class.</p>

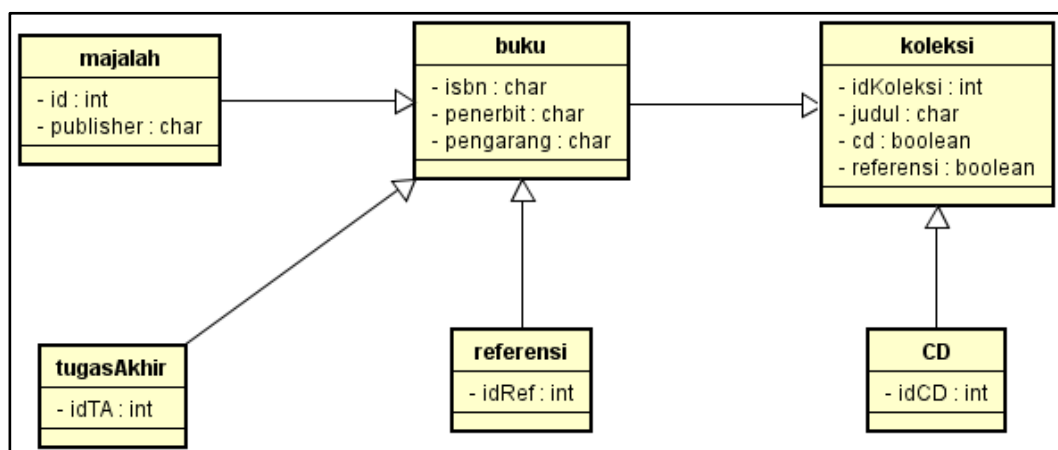


<p><b>Multiplicity</b></p>	<p>Mengindikasikan berapa banyak objek dari suatu kelas terelasi ke objek yang lain</p> <table border="1"> <tr> <th>Multiplicity</th> <th>Arti</th> </tr> <tr> <td>*</td> <td>Banyak</td> </tr> <tr> <td>0</td> <td>Nol</td> </tr> <tr> <td>1</td> <td>Satu (bisa tidak ditulis)</td> </tr> <tr> <td>0..*</td> <td>0 sampai banyak</td> </tr> <tr> <td>1..*</td> <td>1 sampai banyak</td> </tr> <tr> <td>0..1</td> <td>0 atau 1</td> </tr> </table>	Multiplicity	Arti	*	Banyak	0	Nol	1	Satu (bisa tidak ditulis)	0..*	0 sampai banyak	1..*	1 sampai banyak	0..1	0 atau 1	
Multiplicity	Arti															
*	Banyak															
0	Nol															
1	Satu (bisa tidak ditulis)															
0..*	0 sampai banyak															
1..*	1 sampai banyak															
0..1	0 atau 1															
<p><b>Modifier</b></p>	<p>Digunakan untuk menandakan bagaimana kelas lain menggunakan attribute, constructor dan method dari suatu kelas.</p> <table border="1"> <thead> <tr> <th>Notasi</th> <th>Keterangan</th> <th>Kegunaan</th> </tr> </thead> <tbody> <tr> <td>+</td> <td>Membuat visible terhadap kelas lain</td> <td>Visibility public</td> </tr> <tr> <td>#</td> <td>Membuat visible terhadap turunannya</td> <td>Visibility protected</td> </tr> <tr> <td>-</td> <td>Membuat visible hanya dalam 1 kelas</td> <td>Visibility private</td> </tr> <tr> <td>~</td> <td>Membuat visible untuk semua kelas dalam satu paket</td> <td>Visibility package</td> </tr> </tbody> </table>	Notasi	Keterangan	Kegunaan	+	Membuat visible terhadap kelas lain	Visibility public	#	Membuat visible terhadap turunannya	Visibility protected	-	Membuat visible hanya dalam 1 kelas	Visibility private	~	Membuat visible untuk semua kelas dalam satu paket	Visibility package
Notasi	Keterangan	Kegunaan														
+	Membuat visible terhadap kelas lain	Visibility public														
#	Membuat visible terhadap turunannya	Visibility protected														
-	Membuat visible hanya dalam 1 kelas	Visibility private														
~	Membuat visible untuk semua kelas dalam satu paket	Visibility package														

**Gambar 32.** Simbol *class diagram*

**Contoh generalisasi :**

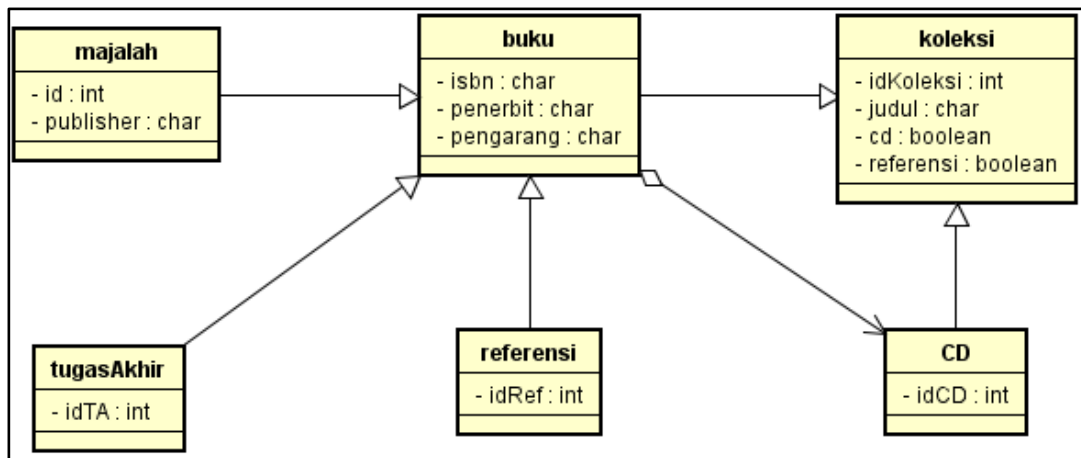
Buku dan CD adalah subclass dari koleksi sedangkan majalah, tugas akhir dan referensi adalah subclass dari buku.



**Gambar 33.** Contoh *generalisasi*

**Contoh agregasi :**

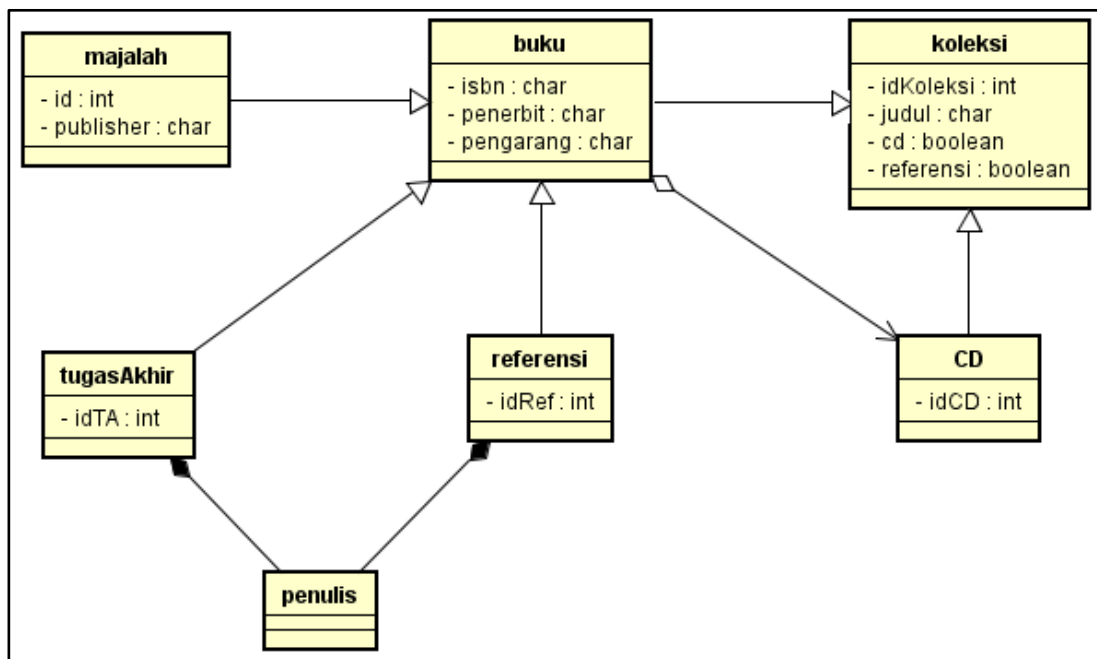
1 buku bisa memiliki 1 CD tetapi boleh juga tidak memiliki CD.



Gambar 34. Contoh agregasi

**Contoh komposisi :**

Relasi antara tugas akhir dan penulis dapat dibaca bahwa sebuah tugas akhir pasti memiliki seorang penulis, demikian juga relasi antara referensi dan penulis.



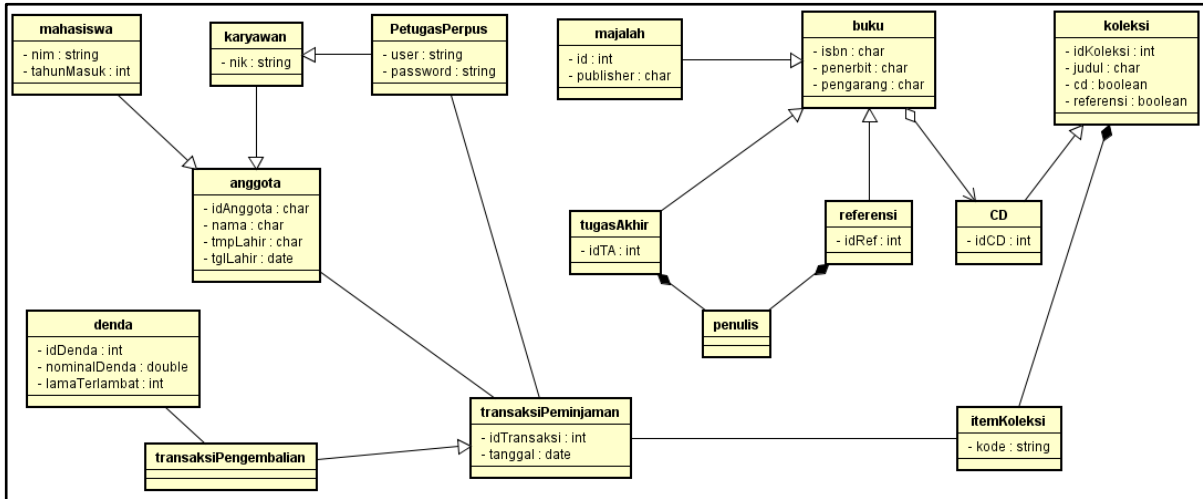
Gambar 35. Contoh komposisi

Langkah – langkah dalam membuat class diagram antara lain :

1. Membuat abstraksi dari suatu kelas
2. Menentukan atribut
3. Menentukan operasi dari kelas tersebut
4. Mengidentifikasi asosiasi dan multiplicity

5. Mengidentifikasi hubungan generalisasi/spesialisasi (jika ada)
6. Mengidentifikasi agregation & dependency

*Class Diagram* di contoh kasus sistem informasi perpustakaan AMIKOM dapat digambarkan dengan :



**Gambar 36.** *Class diagram* sistem informasi perpustakaan AMIKOM

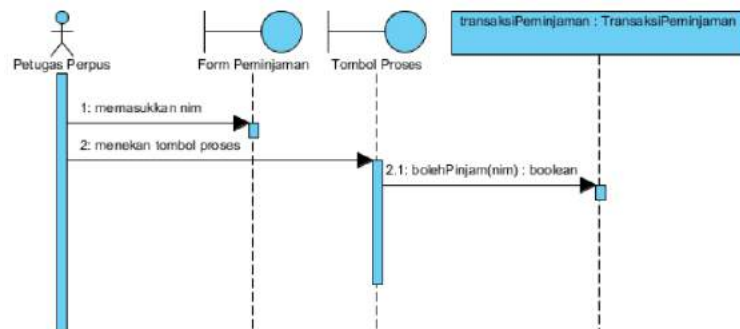
### 7.6 Sequence Diagram

Diagram interaksi atau interaction diagram digunakan untuk memodelkan interaksi objek di dalam sebuah use case (proses). Diagram interaksi memperlihatkan interaksi yang memuat himpunan dari objek dan relasi yang terjadi antar objek tersebut, termasuk juga bagaimana *message* (pesan) mengalir diantar objek.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak use case yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Secara garis besar objek pada *sequence diagram* digambarkan dengan segi empat yang berisi nama dari object yang digarisbawahi. Pada object terdapat 3 cara untuk menamainya yaitu : nama object, nama object dan class serta nama class.

Contoh :


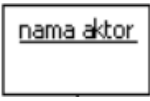
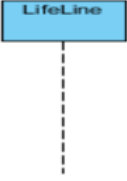
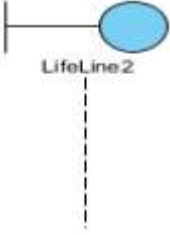



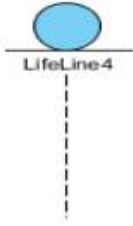


Gambar di atas dapat dibaca : petugas perpustakaan akan memasukkan nim ke form peminjaman. Setelah memasukkan nim ke form peminjaman maka akan menekan tombol proses. Saat tombol proses ditekan maka akan memanggil method bolehpinjam dari objek transaksipeminjaman dan mengisi parameter NIM dari hasil yang didapat dari form peminjaman.

Dalam diagram sequence, setiap object hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar object digambarkan dengan anak panah dari object yang mengirimkan pesan ke object yang menerima pesan.

Berikut adalah simbol-simbol yang ada pada diagram sekuen:

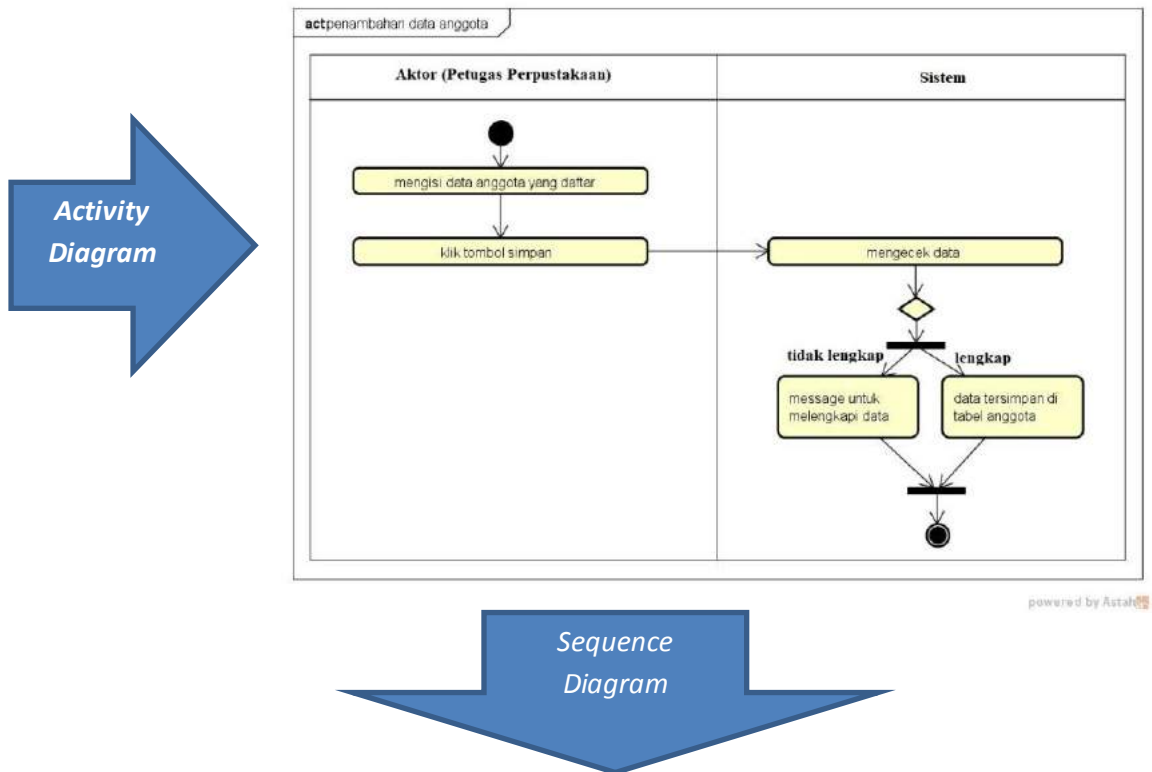
**Tabel 24.** Simbol *sequence diagram*

Simbol	Deskripsi
<p><b>Aktor</b></p>  <p>atau</p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p><b>General Lifeline</b></p> 	<p>Merepresentasikan entitas tunggal dalam sequence diagram, digambarkan dengan kotak. Entitas ini memiliki nama, stereotype atau berupa instance (menggunakan instance:class)</p>
<p><b>Boundary</b></p> 	<p>Boundary biasanya berupa tepi dari system, seperti user interface, atau suatu alat yang berinteraksi dengan system lain.</p>
<p><b>Control</b></p> 	<p>Control element mengatur aliran dari informasi untuk sebuah scenario. Perilaku dan perilaku bisnis umumnya diatur oleh objek ini</p>

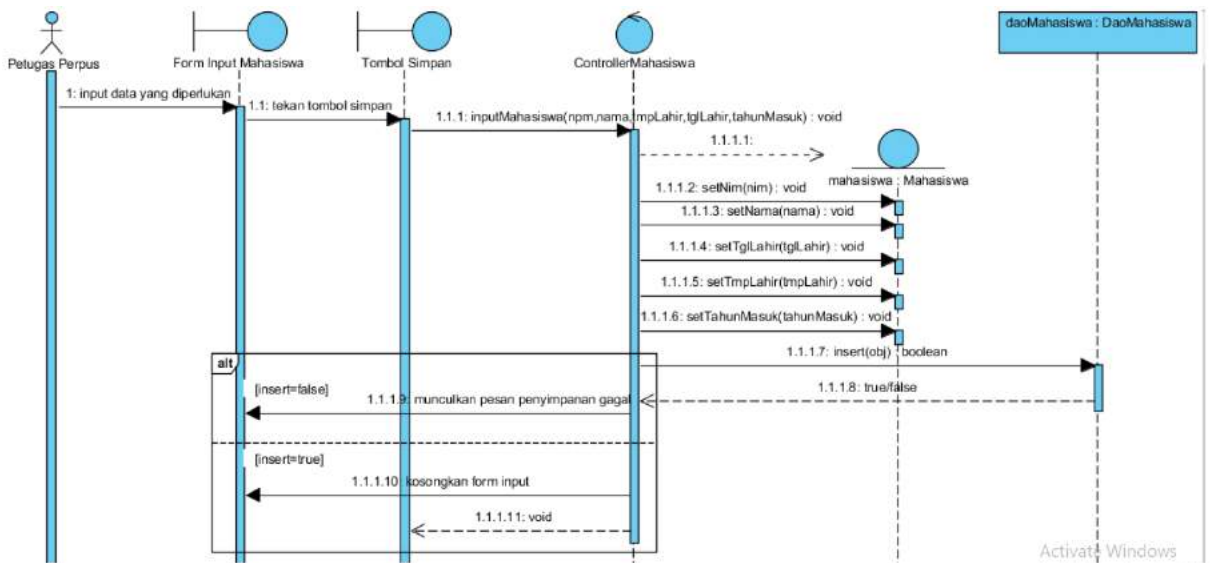
<p><b><u>Entity Lifeline</u></b></p> 	<p>Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object</p>
<p><b><u>Synchronous</u></b></p> 	<p>Relasi ini digunakan untuk memanggil operasi atau method yang dimiliki oleh suatu objek. Synchronous mengharuskan kita menyelesaikan 1 proses baru kemudian memanggil proses berikutnya.</p>
<p><b><u>Asynchronous</u></b></p> 	<p>Relasi ini digunakan untuk memanggil operasi atau method yang dimiliki oleh suatu objek. Asynchronous memberikan kita fasilitas untuk menjalankan proses lain ketika proses sebelumnya belum selesai.</p>
<p><b>Alt</b></p>	<p>Memodelkan if then else blok</p>
<p><b>Opt</b></p>	<p>Memodelkan pernyataan switch</p>
<p><b>Loop</b></p>	<p>Loop fragment</p>

Modul ini hanya mencontohkan 3 activity saja yaitu : penambahan data anggota, peminjaman buku dan pengembalian buku. Untuk lainnya silahkan anda buat :

1. Penambahan data anggota  
Pembuatan activity diagram harus menyesuaikan dengan alur di deskripsi use case input data anggota.



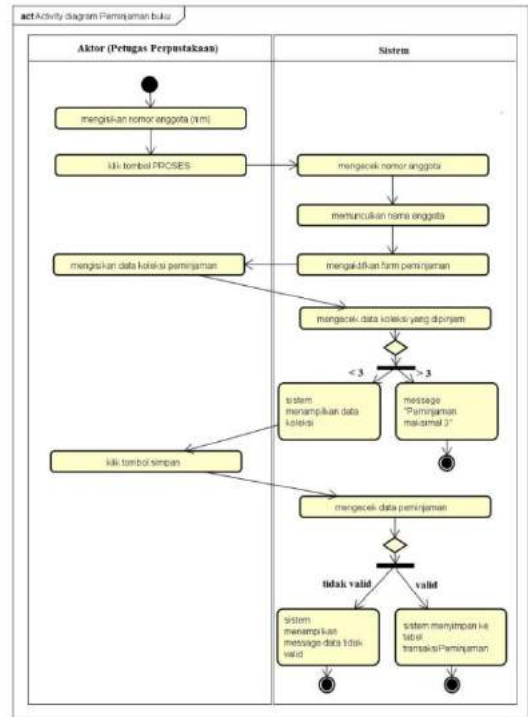
Pembuatan *sequence diagram* harus mengikuti alur *activity diagram* yang telah dibuat.



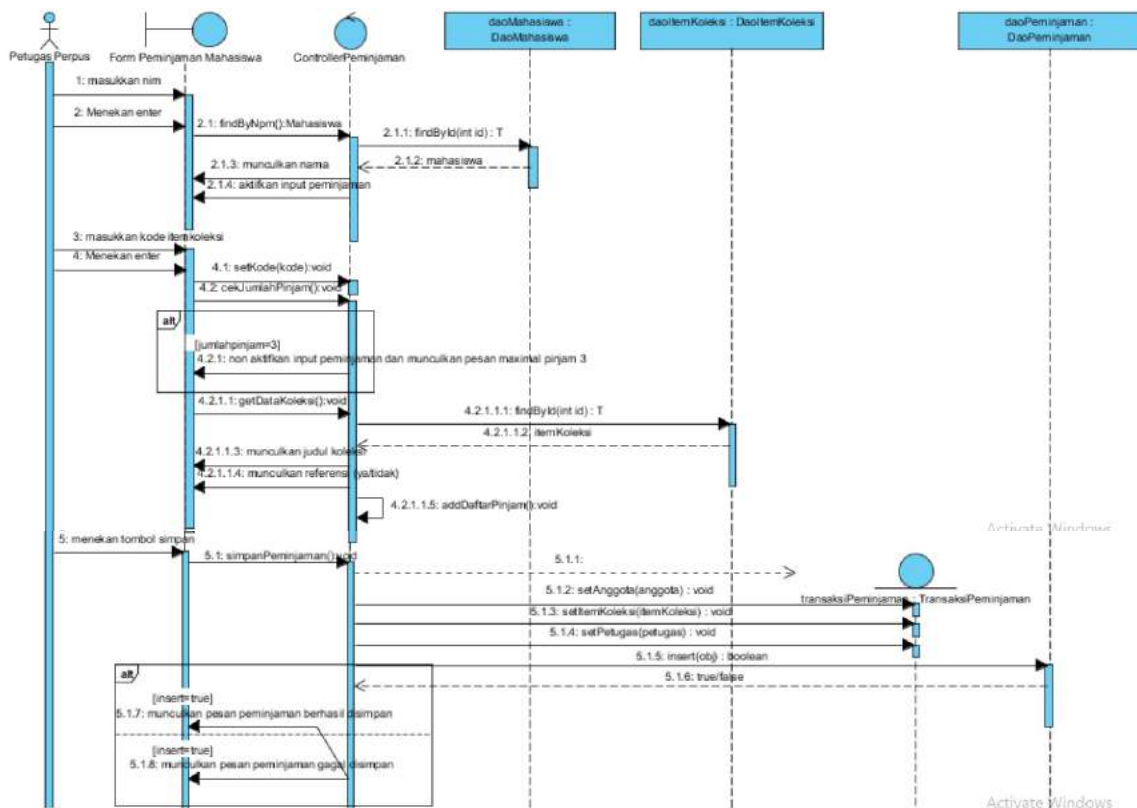
Gambar 37. Sequence diagram pengolahan data anggota

2. Peminjaman buku

Pembuatan *sequence diagram* harus mengikuti alur *activity diagram* yang telah dibuat



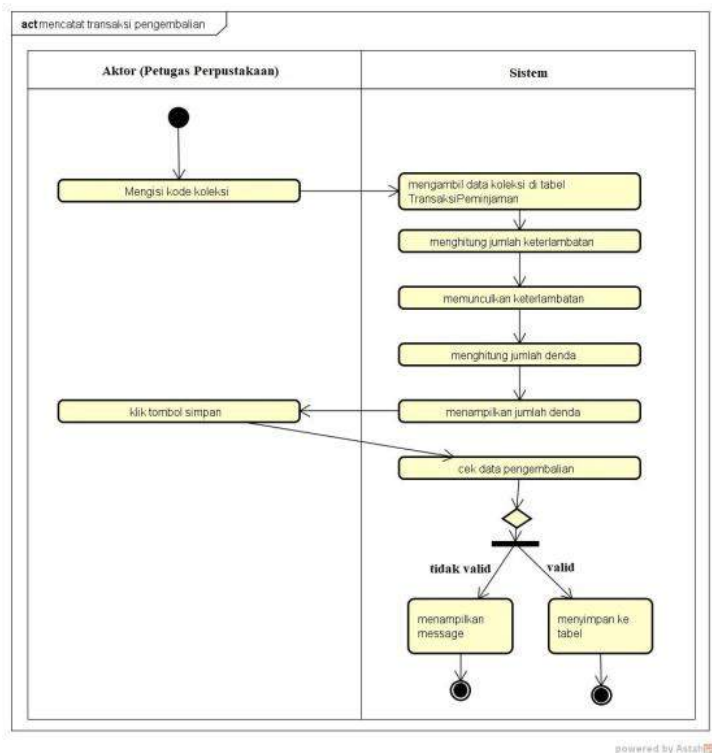
Pembuatan *sequence diagram* harus mengikuti alur *activity diagram* yang telah dibuat.



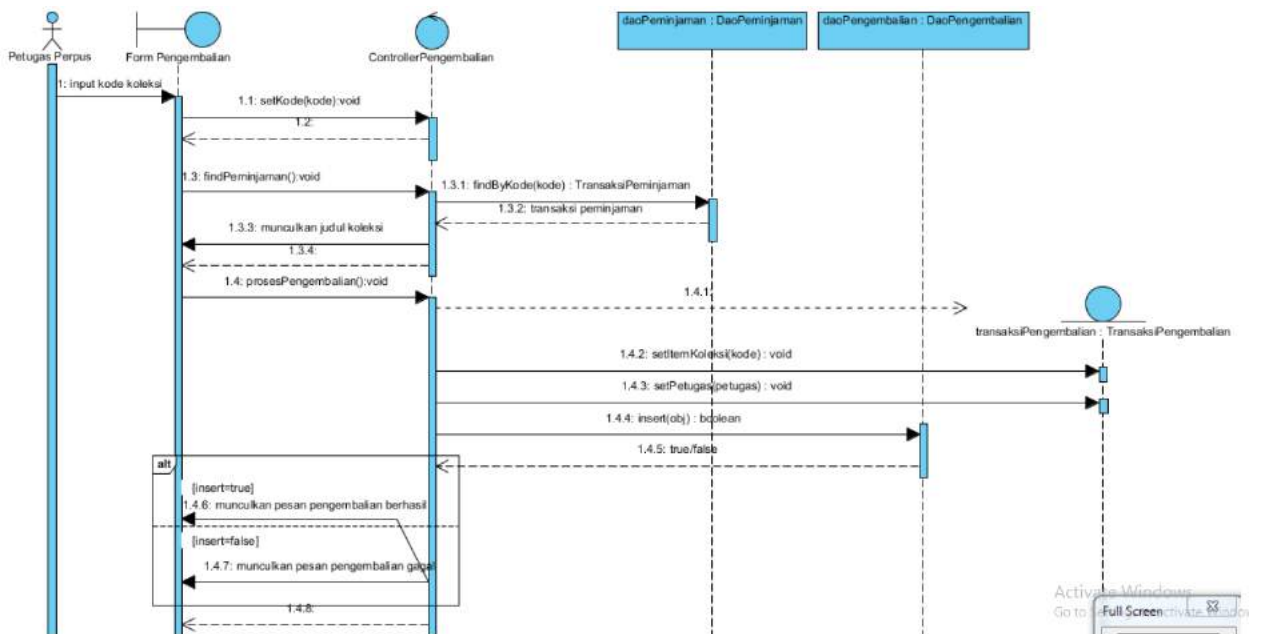
Gambar 38. Sequence diagram mencatat transaksi peminjaman

3. Pengembalian buku

Pembuatan activity diagram harus menyesuaikan dengan alur di deskripsi use case mencatat pengembalian.



Pembuatan *sequence diagram* harus mengikuti alur *activity diagram* yang telah dibuat.



Gambar 39. Sequence diagram mencatat transaksi pengembalian



## **7.7 Tugas**

Buatkan

1. *Use case diagram* dan deskripsi *use case*
2. *Activity diagram*
3. *Class diagram*
4. *Sequence diagram*

Untuk kasus pengembangan sistem informasi yang menjadi tugas kelompok anda !

## BAB VIII PENGUJIAN *BLACK BOX*

---

### Pertemuan ke-24

#### Tujuan :

1. Mahasiswa memahami tahapan-tahapan dalam pengujian sistem
  2. Mahasiswa mampu melakukan pengujian sistem pada kasus yang diangkat
- 

### 8.1 Pengertian *Black Box Testing*

*Black-box testing* adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus (Whitten, 2004).

Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari ke banyak jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga. Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari software. Ujicoba *blackbox* merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya. Ujicoba *blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya (Whitten, 2004):

- a. Fungsi-fungsi yang salah atau hilang
- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja
- e. Kesalahan Inisialisasi dan terminasi

#### Ciri-Ciri *Black Box Testing* [1] :

1. *Black box testing* berfokus pada kebutuhan fungsional pada software, berdasarkan pada spesifikasi kebutuhan dari software.
2. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*.

Berikut adalah pengujian *Black Box Testing* untuk kasus sistem informasi Perpustakaan AMIKOM

**Tabel 25.** Pengujian *Black Box testing* sistem informasi Perpustakaan AMIKOM

Nomor Pengujian	Deskripsi	Prosedur Pengujian	Masukkan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang Didapat	Kesimpulan
1	Pengolahan data transaksi peminjaman	Menginputkan data peminjaman buku	Semua text di form peminjaman buku diisi	Buku yang dipinjam dapat terinput dan tersimpan di tabel transaksiPeminjaman	Data yang diinputkan tersimpan di tabel transaksiPeminjaman	Data yang diinputkan tersimpan di tabel transaksiPeminjaman	Diterima
			Beberapa isian di form peminjaman buku dikosongkan	Muncul message box yang berisi peringatan	Muncul pesan error dan tidak tersimpan di tabel transaksiPeminjaman	Message box yangberisi peringatan tidak muncul dan data tersimpan di tabel transaksiPeminjaman	Tidak diterima
			Anggota berstatus mahasiswa meminjam buku lebh dari 3	Aplikasi akan menolak melakukan penginputan data transaksi peminjaman baru jika diketahui ada anggota yang berstatus mahasiswa, meminjam buku lebih dari 3 buah	Muncul pesan error dan tidak tersimpan di tabel transaksiPeminjaman	Message box yang berisi peringatan tidak boleh meminjam lebih dari 3 buku muncul dan data tidak tersimpan di tabel transaksiPeminjaman	Diterima

2	Petugas memiliki hak akses untuk melakukan pengolahan data pengembalian buku	Menginputkan data pengembalian buku	Semua text di form pengembalian buku diisi	Buku yang dikembalikan dapat terinput dan tersimpan di tabel transaksiPengembalian	Data yang diinputkan tersimpan di tabel transaksiPengembalian	Data yang diinputkan tersimpan di tabel transaksiPengembalian	Diterima
			Beberapa isian di form pengembalian buku dikosongkan	Muncul message box yang berisi peringatan	Muncul pesan error dan tidak tersimpan di tabel transaksiPengembalian	Message box yang berisi peringatan muncul dan data tidak tersimpan di tabel transaksiPengembalian	Diterima
			Ketika buku dikembalikan, sistem secara otomatis melakukan pengecekan keterlambatan buku	Muncul message box yang berisi nominal denda	Informasi denda terinput ke tabel denda	Tidak muncul message box yang berisi nominal denda	Tidak diterima
....	....	....	....	....	....	....	....
<b>Catatan : Semua kebutuhan fungsional dan nonfungsional harus diujikan</b>							

## 8.2 Tugas

Buatlah pengujian *Black Box testing* untuk tugas yang sedang anda kerjakan !

## DAFTAR PUSTAKA

- [1] Whitten, J.L., Bentley, L.D., dan Dittman, K.C., 2004, *Metode Desain dan Analisis Sistem*, McGraw-Hill, disadur oleh Andi Offset.
- [2] O'Brien, J.A., dan Marakas, G.M., 2011, *Management Information Systems 10 edition*, McGraw-Hill Education.
- [3] Leitch and Robert A, 1983, *Accounting Information System*, Prentice Hall
- [4] Davis, G dan Olson, M., 2012, *Management Information Systems 2nd edition*, Tata Mcgraw Hill.
- [5] Stair, R dan Reynolds,G., 2017, *Fundamentals of Information Systems 9th Edition*, Cengage Learning.
- [6] Laudon, K.C dan Laudon, J.P., 2015, *Management Information Systems: Managing the Digital Firm, Student Value Edition (14th Edition)*, Pearson.
- [7] McLeod,R dan Schell,G., 2006, *Management Information Systems (10th Edition)*, Pearson.
- [8] Turban,E., Aronson,J.E., dan Liang,T., 2007, *Decision Support Systems and Intelligent Systems*, Andi Offset.
- [9] Al Fatta, H., 2007, *Analisa dan Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*, Andi.
- [10] Indrajit, R.E., 1999, Sistem Informasi Eksekutif, diakses di : <http://www.indrajit.org>.
- [11] Pratama, A.A., Alamsyah, A., Sari, P.K., *Studi Kelayakan Sistem Penilaian Essay Dalam Bahasa Inggris Secara Otomatis*, Jurnal Prodi S1 Manajemen Bisnis Telekomunikasi dan Informatika, Fakultas Ekonomi dan Bisnis, Universitas Telkom.
- [12] Nugroho, A., 2009, *Rekayasa Perangkat Lunak Menggunakan UML dan Java*, Andi.
- [13] Fowler, M., 2004, *UML Distilled A Brief Guide To The Standard Object Modelling Language*, Addison-Wesley
- [14] Admin, Java Bootcamp 2011, STMIK AMIKOM Yogyakarta