# PRAKATA

Puji syukur Alhamdulillah, penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan Rahmat dan Karunia Nya, sehingga pada akhirnya penulis dapat menyelesaikan buku ini dengan baik. Dimana buku ini penulis sajikan dalam bentuk buku yang sederhana. Adapun buku ini penulis buat untuk menambah wawasan para pembaca pada umumnya dan untuk menambah bahan materi untuk mata kuliah Analisa Perancangan Sistem Informasi bagi mahasiswa pada khususnya.

Sebagai bahan penulisan diambil berdasarkan pencarian di beberapa sumber, seperti buku, internet dan masih banyak lagi yang lainnya. Dalam buku ini dijelaskan bagiamana sebuah sistem informasi dibuat dengan tahapan-tahapan yang sesuai dengan metode yang ada. Penulis menyadari bahwa tanpa bimbingan dan dorongan dari semua pihak, maka penulisan dan pembuatan buku ini ini tidak akan berjalan dengan lancar.

Penulis mengucapkan terima kasih kepada tim sehingga bisa menyelesaikan penyusunan buku ini. Semoga buku ini berguna bagi para pembaca baik mahasiswa ataupun siapapun yang bisa dijadikan bahan referensi untuk pembelajaran.

Jakarta, 15 Maret 2019

Penulis

# DAFTAR ISI

[PRAKATA 2](#_Toc3368871)

[DAFTAR ISI 3](#_Toc3368872)

[BAB I PENELITIAN 5](#_Toc3368873)

[1.1. Pengertian Umum 5](#_Toc3368874)

[1.2. Penelitian Dibidang Ilmu Komputer 6](#_Toc3368875)

[1.3. Latar Belakang Masalah Penelitian 8](#_Toc3368876)

[1.4. Rumusan Dan Pembatasan Masalah Penelitian 9](#_Toc3368877)

[1.5. Panduan Penggunaan Software Manajemen Referensi Mendeley 12](#_Toc3368878)

[1.6. Latihan 13](#_Toc3368879)

[BAB II PERUMUSAN MASALAH DAN ANALISA DOKUMEN I/O SYSTEM BERJALAN 15](#_Toc3368880)

[2.1. Metode Pengembangan Sistem 15](#_Toc3368881)

[2.2. Teknik Pengembangan Sistem 17](#_Toc3368882)

[BAB III ANALISA SISTEM INFORMASI 23](#_Toc3368883)

[3.1. Analisa Kebutuhan Sistem 23](#_Toc3368884)

[3.2. Metode Pengumpulan Data 24](#_Toc3368885)

[3.3. Perumusan Proses Bisnis 25](#_Toc3368886)

[3.4. Penggambaran Activity Diagram Sistem Berjalan 26](#_Toc3368887)

[BAB IV DESAIN DATABASE & STRUKTUR KODE 29](#_Toc3368888)

[4.1. Pengertian ERD 29](#_Toc3368889)

[4.2. Simbol pada ERD 30](#_Toc3368890)

[4.3. Tahapan Pembuatan & Contoh ERD 34](#_Toc3368891)

[4.4. Spesifikasi File 36](#_Toc3368892)

[4.5. Struktur Kode 41](#_Toc3368893)

[BAB V *UNIFIED MODELING LANGUAGE* 43](#_Toc3368894)

[5.1 PENGERTIAN *UNIFIED MODELING LANGUAGE* (UML) 43](#_Toc3368895)

[A. USE CASE DIAGRAM 44](#_Toc3368896)

[B. ACTIVITY DIAGRAM 46](#_Toc3368897)

[C. CLASS DIAGRAM 51](#_Toc3368898)

[D. SEQUENCE DIAGRAM 57](#_Toc3368899)

[5.2 STUDI KASUS 61](#_Toc3368900)

[BAB VI DESAIN INTERFACE 67](#_Toc3368901)

[6.1 ANALISA PERANCANGAN INPUT & OUTPUT 67](#_Toc3368902)

[A. Perancangan Output 68](#_Toc3368903)

[B. Perancangan Input 71](#_Toc3368904)

[C. Merancang Arsitektur Antarmuka 73](#_Toc3368905)

[6.2. STUDI KASUS RANCANGAN DOKUMEN INPUT & OUTPUT 74](#_Toc3368906)

[BAB VII PENGUJIAN PERANGKAT LUNAK DAN PEMELIHARAAN SISTEM 77](#_Toc3368907)

[7.1. Pengujian Perangkat Lunak 77](#_Toc3368908)

[7.2. Pemeliharaan Sistem 84](#_Toc3368909)

[DAFTAR PUSTAKA 87](#_Toc3368910)

[BIODATA 88](#_Toc3368911)

# BAB I PENELITIAN

**Deskripsi :**

Membahas pengertian umum tentang penelitian, memahami penelitian bidang ilmu sistem informasi, memahami teknik pembahasan pada latar belakang masalah, mampu menggunakan aplikasi mendelay da lam mengutip sebagai penguat argumentasi dari suatu masalah, melatih menemukan permasalahan dari beberapa kasus yang disajikan.

**Tujuan Pembelajaran:**

Setelah melakukan bagian ini mahasiswa mampu :

* + - 1. Memahami konsep penelitian dibidang ilmu sistem informasi
      2. Mampu melihat dan menemukan permasalahan
      3. Mampu menyusun lata belakang masalah
      4. Mampu menggunakan aplikasi mendelay dalam menyertakan kutipan dalam penelitian

## Pengertian Umum

Berawal dari suatu pertanyaan sederhana dikalangan mahasiswa, Apa itu penelitian?. Beberapa pandangan terkait penelitian, antara lain

1. Penelitian dilakukan karena ada masalah penelitian, dimana masalah penelitian sendiri muncul karena ada latar belakang masalah.
2. Penelitian dilakukan secara terencana, sistematis, berulang-ulang dan terukur
3. Penelitian harus memiliki orisinalitas dan kebaruan, serta menghasilkan kontribusi yang orisinil pada pengetahuan dalam bentuk menemukan dan merevisi fakta, teori dan aplikasi.

Dari ketiga pandangan di atas, dapat disimpulkan bahwa penelitian adalah sebagai media atau penyaluran keingginan manusia untuk mengetahui terhadap suatu masalah dengan melakukan tindakan atau perlakuan seperti memeriksa, menganalisa, membandingkan, mengusut, menelaah dan mempelajari secara sistematis dan sunguh-sunguh). Lebih sederhana lagi penelitian mengandung beberapa komponen, yaitu:

* 1. Ada rasa ingin tahu
  2. Ada sesuatu atau masalah
  3. Ada proses atau usaha untuk menyelesaikan sesuatu atau masalah tersebut
  4. Ada hasilnya atau karya

Masalah sebagai kunci utama dalam melakukan penelitian, dari masalah yang ada dilakukan penelitian untuk mendapatkan berbagai solusi melalui metode penyelesaian yang akan dipergunakan.

## Penelitian Dibidang Ilmu Komputer

Pada bidang ilmu komputer, luaran penelitian dapat berbentuk karya terapan (*software*) yang siap diimplementasikan, layak untuk diajukan untuk hak paten dan paper yang dihasilkan dapat dipublikasikan di berbagai media jurnal ilmiah. Merujuk pada Buku Panduan Program Kreatifitas Mahasiswa (PKM) yang diterbitkan Direktorat Jenderal Pembelajaran Dan Kemahasiswaan Kementrian Riset, Teknologi, Dan Pendidikan Tinggi terdapat beberapa penelitian yang memiliki hubungan erat dengan bidang ilmu komputer, yaitu:

Tabel 1. Kriteria Program Kreatifitas Mahasiswa(PKM)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Kriteria | Bidang Kegiatan | | | | | | |
| PKM-P | PKM-K | PKM-M | PKM-T | PKM-KC | PKM-AI | PKM-GT |
| 1 | Inti Kegiatan | Karya kreatif, inovatif dalam penelitian | Karya kreatif, inovatif dalam membuka peluang usaha bagi mahasiswa | Karya kreatif, inovatif dalam mem bantu masyarakat | Karya kreatif, inovatif dalam menciptaka n karya teknologi | Karya kreatif, inovatif dalam Iptek | Karya kreatif, dalam penulisan artikel ilmiah | Karya tulis dalam penuangan gagasan/ ide kreatif |
| 2 | Materi Kegiatan | Sesuai bidangilmu, lintas bidang di anjurkan | Semua bidang ilmu atau yang relevan | Semua bidang ilmu atau yang relevan | Semua bidang ilmu, lintas bidang dianjurkan | Semua bidang ilmu atau yang relevan | Karya kelompok yang telah dilaksanakan | Karya kelompok |
| 3 | Strata Pendidikan | Diploma, S-1 | Diploma, S-1 | Diploma, S-1 | Diploma, S-1 | Diploma, S-1 | Diploma, S-1 | Diploma, S-1 |
| 4 | Jumlah Anggota | 3 orang | 3-5 orang | 3-5 orang | 3-5 orang | 3 orang | 3 orang | 3 orang |
| 5 | Laporan Akhir | Hasil Kerja | Hasil Kerja | Hasil Kerja | Hasil Kerja | Hasil Kerja | Hasil Kerja | Hasil Kerja |
| 6 | Luaran | Artikel, potensi paten | Barang dan jasa komersial dan artikel | Jasa, desain, barang dan artikel | Model desain, piranti lunak, jasa, Artikel, dan potensi paten | Sistem, desain, barang, prototip dan artikel | Artikel Ilmiah | Gagasan kreatif yang tertulisdan artikel. |

Komparasi penelitian jenjang pendidikan D3/D4 vs S1 vs S2 vs S3

D3/D4

1. Pengembangan Sistem Informasi Rumah Sakit untuk Rumah Sakit “XYZ”
2. Karakter: menguasai skill teknis
3. S1
4. Sistem cerdas berbasis Neural Network untuk Prediksi Harga Saham
5. Karakter: menguji teori, ada software development
6. S2/S3
7. Penerapan Algoritma Genetika untuk Pemilihan Arsitektur Jaringan Secara Otomatis pada Neural Network untuk prediksi Harga Saham
8. Karakter: mengembangkan teori (perbaikan metode), ada kontribusi ke teori/metode.

## Latar Belakang Masalah Penelitian

Secara umum, pentingnya penelitian dapat disebabkan karena adanya sesuatu hal, yaitu:

1. Fenomena tertentu yang menunjukan adanya celah (gap) antara fenomena yang terjadi dengan yang diharapkan dalam bisnis, masyarakat, industri, atau negara yang menarik perhatian peneliti.
2. Celah (gap) hasil penelitian menunjukan adanya perbedaan hasil-hasil penelitan yang dilakukan oleh peneliti tertentu dengan peneliti lainnya.
3. Celah (gap) teori yang menunjukan adanya perbedaan pandangan teori antara ahli tertentu dengan ahli lainnya.

Masalah adalah penyimpangan antara yang diharapkan dengan kejadian atau kenyataan dan masalah tersebut dapat diselesaikan. Masalah penelitian adalah masalah yang akan menjadi obyek penelitian. Suatu masalah dapat dijadikan masalah penelitian apabila :

1. Masalah tersebut dengan observasi atau pengumpulan data dapat memberi jawaban,
2. Nilai yang disertakan dalam masalah yang diamati dapat diukur.

Menurut Nazir dalam (Limbong, 2012) ciri-ciri masalah yang baik adalah masalah yang dipilih harus mempunyai nilai penelitian:

Masalah harus mempunyai keaslian/menyangkut hal-hal yang up to date dan baru, mempunyai nilai ilmiah atau aplikasi ilmiah.

Masalah harus menyatakan suatu hubungan.

Masalah harus dapat diuji dengan perlakukan-perlakuan serta data dan fasilitas yang ada.

Masalah harus mempunyai arti dan nilai baik dalam bidang ilmunya sendiri maupun dalam bidang aplikasi untuk penelitian terapan.

Masalah harus dapat dinyatakan dalam bentuk pertanyaan yang jelas dan tidak membingungkan.

Masalah yang dipilih harus mempunyai fisibilitas atau dapat dipecahkan

Masalah yang dipilih harus sesuai dengan kualifikasi si peneliti

Masalah untuk penelitian dapat diperoleh antara lain dari :

1. Pengamatan terhadap kegiatan manusia
2. Pengamatan terhadap alam sekeliling
3. Bacaan
4. Ulangan serta perluasan penelitian
5. Cabang studi yang sedang dikembangkan
6. Pengalaman dan catatan pribadi
7. Praktek serta keinginan masyarakat
8. Bidang spesialisasi
9. Pelajaran atau mata ajaran yang sedang diikuti
10. Analisis bidang pengetahuan
11. Diskusi-diskusi ilmiah

Menurut (Antonius, 2010), latar belakang masalah berisi uraian antara lain:

1. Memaparkan pentingnya masalah yang akan diteliti
2. Alasan-alasan pemilihan masalah, paling sedikit 3(tiga) poin
3. Hubungan antara masalah ilmiah yang akan diteliti dan kepentingan umum, pemerintah, dan sebagainya.

## 1.4. Rumusan Dan Pembatasan Masalah Penelitian

**A. Rumusan Masalah Penelitian**

F.N. Kerlinger dan H.B. Lee (2009) serta Uma Sekaran dan Roger Bougie (2010) dalam (Sugiarto, 2017) berpendapat bahwa kriteria umum yang harus dipertimbangkan oleh seorang peneliti dalam menentukan dan merumuskan masalah penelitian, antara lain:

1. Masalah yang akan diteliti harus memiliki teoritis maupun implikasi praktik
2. Masalah yang diteliti menarik bagi peneliti dan sesuai dengan bidang ilmu yang dikuasainya.

Rumusan masalah adalah pernyataan singkat suatu masalah yang akan diteliti. Lebih baik dirumuskan dengan prinsip 5W + 1 H. Cara membuat rumusan masalah adalah sebagai berikut :

1. Rumusan masalah dinyatakan dalam bentuk pertanyaan
2. Rumusan masalah hendaknya jelas dan padat
3. Rumusan masalah harus berisi implikasi adanya data untuk memecahkan masalah
4. Rumusan masalah harus merupakan dasar dalam membuat hipotesis

Menurut (Antonius, 2010) perumusan masalah memiliki ciri sebagai berikut:

1. Masalah penenelitian dirumuskan dalam kalimat tanya
2. Variabel yang dimunculkan dalam rumusan masalah harus dapat terukur dalam proses penelitian, baik secara kuantitatif maupun kualitatif.
3. Kata kunci dalam merumuskan masalah, antara lain:
4. Adakah …………………………….?
5. Apakah ……………………………?
6. Bagaimana pengaruh ……………..?
7. Sejauh manakah……………………?
8. Tetapkan satu atau lebih kata kuncinya, sesuai dengan ruang lingkup permasalahan atau penelitian.

Contoh 1:

Tabel 2. Contoh Rumusan Masalah 1

|  |  |
| --- | --- |
| Judul Penelitian | Rumusan Masalah |
| Analisis pengaruh pelatihan terhadap kemampuan membaca berita karyawan TVRI stasiun pusat Jakarta tahun 2000 | 1. Bagaimana bentuk hubungan antara pelatihan dan kemampuan membaca berita? 2. Seberapa besar pengaruh pelatihan terhadap kemampuan membaca berita ? |

Contoh 2 :

Tabel 3. Contoh Rumusan Masalah 2

|  |  |
| --- | --- |
| Judul Penelitian | Rumusan Masalah |
| Aplikasi Pengamanan Pengiriman Pesan Teks Digital Pada Jaringan Lan Dengan Metode Message Digest 5 (MD5) | 1. Bagaimana teknik untuk mengamankan pesan teks digital? 2. Bagaimana menerapkan metode message digest 5 untuk mengamankan pengiriman pesan teks digital pada jaringan LAN ? 3. Bagaimana merancang Aplikasi Pengamanan Pengiriman Pesan Teks Digital Pada Jaringan LAN dengan Metode Message Digest 5 (MD5)? |

* + 1. **Pembatasan Masalah Penelitian**

Menurut (Hasibuan , 2017), Masalah dan Pembatasan Masalah Memberikan batasan yang jelas bagian mana dari persoalan yang dikaji dan bagian mana yang tidak.

Dalam melaksanakan penelitian, diperlukan ketegasan untuk menyatakan batasan masalah secara eksplisit, terutama bila masalah yang diteliti terlalu luas sementara peneliti hanya memfokuskan perhatiannya pada sebagian aspek dari masalah tersebut. Pembatasan penelitian menampilkan variabel-variabel, faktor-faktor, dan menetapkan batasan yang jelas tentang tempat dan waktu diadakan penelitian. Tujuan pembatasan masalah penelitian:

1. Agar masalah penelitian lebih fokus, mudah dirumuskan dan tidak melebar kemana-mana dan tidak membingungkan para pembacanya.
2. Agar pembahasan lebih terperinci dan memungkinkan pengambilan keputusan yang definitif.

Penelitian Ilmiah dalam pelaksanaannya menggunakan kaidah-kaidah ilmiah, salah satu ciri-cirinya adalah *Generalizability*, artinya semakin luas ruang lingkup penggunaan hasil penelitian semakin berguna (Sesuaikan dengan Kemampuan dan waktu diharapkan ada Batasan Masalah yang jelas).

## Panduan Penggunaan Software Manajemen Referensi Mendeley

Menurut LestariMendeley adalah software manajemen referensi dan jaringan sosial akademis yang bisa membantu Anda mengorganisir penelitian, berkolaborasi dengan peneliti lain secara online dan menemukan publikasi penelitian terakhir. Sebagai database referensi, file referensi seperti buku atau artikel dari jurnal dalam bentuk PDF bisa disimpan dan diberi keterangan yang tepat untuk membantu mempermudah pencarian. File-file PDF yang disimpan juga bisa dibuka, dibaca, dan diberi catatancatatan dengan sticky notes atau highlight. Tulisan yang dibuat dengan Microsoft Word, OpenOffice atau LaTex bisa dihubungkan dengan software Mendeley sehingga sitasi dan daftar referensi (bibliography) bisa disusun secara otomatis. Mendeley juga bisa dihubungkan dengan software manajemen referensi lainnya seperti EndNote, Papers dan Zotero. Setiap instalasi Mendeley harus disertai dengan pendaftaran akun baru, karena setiap akun Mendeley disertai dengan akun online. Jika pengguna Mendeley mengsinkronisasi data-data PDF yang disimpan di komputer/laptop dengan akun onlinenya, maka informasi referensi yang disimpan di laptop tadi juga akan tersimpan di website dan bisa diakses dari manapun via internet di laptop atau dengan aplikasi Mendeley untuk iPhone dan iPad. Selain itu melalui jaringan internet, bisa ditemukan peneliti atau kelompok peneliti/penulis lain yang memiliki kesamaan minat atau melakukan pencarian artikel-artikel yang sehubungan dengan topik penelitian yang sedang dikerjakan. Lebih lengkap panduan mendeley dapat diunduh melalui link sebagai berikut:

[**https://drive.google.com/file/d/1jGLJ9JgHyzcKAxlcy08xMkIesVrjTPxp/view?usp=sharing**](https://drive.google.com/file/d/1jGLJ9JgHyzcKAxlcy08xMkIesVrjTPxp/view?usp=sharing)

## Latihan

Lulusan Perguruan Tinggi dituntut untuk memiliki *academic knowledge*, *skill of thinking*, *management skill, dan communication skill.* Untuk meningkatkan kemampuan tersebut penelitian bagian dari mata kuliah yang dapat dijadikan sebagai media untuk mewujudkan kemampuan tersebut dengan disajikan berbagai latihan studi kasus. Untuk meningkatkan mahasiswa memiliki kemamampuan menemukan berbagai permasalahan dilingkungan masyarakat, metode yang dipergunakan dalam kasus ini adalah pembuktian terbalik, artinya soal studi kasus akan diambil dari apa yang sudah diimplementasikan dan mahasiswa memberikan uraian permasalahan yang ada dan solusi yang diberikan:

Studi Kasus 1:

1. Sebagai pengguna jalan raya, pasti akan menemukan banyak pertigaan, perempatan yang telah dipasang *traffig light*
2. Instruksi tugas sebagai berikut:

Uraikan dengan jelas dan rinci masalah yang ditemukan apabila pertigaan, perempatan dijalan raya tidak dilengkapi dengan *traffic light*

Uraikan dengan jelas dan rinci apakah adanya *traffic light* mampu menyelesaikan masalah dari point a?

Dengan adanya teknologi informasi *traffic light* di pertigaan dan perempatan sebutkan dampak yang di rasakan oleh:

1. Pengguna jalan raya
2. Petugas Dinas Lalu Lintas Angkutan Jalan (DLLAJ)

Studi kasus 2:

1. Dengan meningkatnya ekonomi masyarakat, setiap rumah sudah memiliki televisi dengan berbagai merk. Teknologi televisi telah dilengkapi dengan remote kontrol yang dapat dipergunakan oleh pemilik televisi, misal mengurangi volume, mengganti chanel dan lain-lain
2. Instruksi tugas sebagai berikut:
3. Sebutkan permasalahan yang diakibatkan, apabila televisi anda tidak memiliki tools tambahan seperti remote kontrol
4. Sebutkan kenyamanan anda mengoperasikan televisi menggunakan remote kontrol?

**Keterangan:** Tugas diatas sebagai latihan bagaimana menemukan permasalahan yang ada di kehidupan masyarakat dan solusi yang diberikan.

# BAB II PERUMUSAN MASALAH DAN ANALISA DOKUMEN I/O SYSTEM BERJALAN

**Deskripsi :**

Membahas materi tentang metode pengembangan system (*waterfall*), perumusan masalah, dan analisa dokumen i/o sistem yang berjalan.

**Tujuan Pembelajaran:**

Mahasiswa paham tentang Metode Pengembangan system, merumuskan masalah dalam sebuah system, menganalisa dokumen input dan output pada system berjalan.

## 2.1. Metode Pengembangan Sistem

Sistem yang sedang berjalan pada suatu organisasi akan terus dikembangkan dan dilakukan penyempurnaan untuk memperbaiki kekurangan yang terdapat pada sistem yang berjalan. SDLC atau *Software Development Life Cycle* atau *System Development Life Cycle* adalah sebuah metode yang digunakan untuk mengembangkan sistem. SDLC adalah sebuah proses logika yang digunakan oleh seorang system analist untuk mengembangkan sebuah sistem informasi yang melibatkan requirements, validation, training dan pemilik (Mulyani, 2016)

Tiga langkah umum yang dilalui pada proses perancangan sistem :  
1.    Mendefinisikan Sistem Informasi yang dibangun

2.    Memilih metode Pengembangan Sistem Informasi

3.    Memilih Teknik Perancangan Arsitektur Sistem Informasi

SDLC identik dengan teknik pengembangan sistem waterfall, dikarenakan tahapannya memurun dari atas kebawah. Tahapan SDLC :

1. Planning
2. Analysist
3. Design
4. Implementation
5. Use

Tahapan pengembangan sistem SDLC Waterfall (Sukamto, 2014):

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user.* Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

Misal kebutuhan (*system requirement*) dari sistem penyewaan.

1. Halaman *User yaitu* Pengunjung dapat melihat informasi website, Pengunjung melakukan registrasi dan menjadi member, Member dapat melihat daftar pemesanan lapangan, Member dapat *login* dengan *account* yang telah dibuat, Member dapat melakukan pemesanan secara *online,* Member dapat melakukan konfirmasi pembayaran, Member dapat mencetak bukti pemesanan
2. Sedangkan halaman Admin misalnya Admin dapat mengelola data member, Admin dapat mengelola harga lapangan, Admin dapat mengelola data galeri, Admin dapat mengelola data data berita, Admin dapat megelola data transaksi, Admin dapat mengelola data laporan
3. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak representasi antar muka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

Contoh: *Unified Modeling Language* (UML) untuk merancang dan mendokumentasikan sistem yang dibuat. Sedangkan untuk menggambarkan relasi antara objek dapat menggunakan *Entity Relationship Diagram* (ERD).

1. Pembuatan Pengkodean Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

Contoh :bahasa pemrograman PHP dengan suport database menggunakan MySql .

1. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

contoh: *Black box* *testing* dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak

1. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terditeksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

Contoh: software maupun hardware yang sudah ditentukan baik pada saat pembuatan maupun operasi program.

## 2.2. Teknik Pengembangan Sistem

Menurut (Sulianta, 2017) ada tiga langkah umum yang dilalui pada proses perancangan:

1. Mendefinisikan Sistem Informasi yang dibangun
2. Memilih metode Pengembangan Sistem Informasi
3. Memilih Teknik Perancangan Arsitektur Sistem Informasi

Perkembangan teknologi membuat sistem yang digunakanpun semakin kompleks sedangkan waktu yang dibutuhkan terbatas, sehingga menyembabkan sistem SDLC yang dirasa saat ini tidak cukup memberikan solusi sehingga teknik-teknik pengembangan sistem yang merupakan pengembangan dari metode SDLC yaitu:

1. Prototyping
2. Rapid Application Development (RAD)
3. Iteratif
4. Spiral
5. Joint Application Development (JAD)
6. Unified Modelling Language (UML)
7. **Prototyping**

Prototyping merupakan teknik pengembangan sistem yang menggunakan prototype untuk menggambarkan sistem, sehingga pengguna atau pemilik mempunyai gambaran tentang sistem yang akan dilakukan (Mulyani, 2016). Dalam penggambaran sistem informasi protorype diwujudkan dalam bentuk user interface program aplikasi dan contoh-contoh reporting yang dihasilkan, sehingga pengguna akan mempunyai gambaran tentang sistem yang akan digunakan nanti.

(Mc Leod & Schell, 2007) mendifinisikan 2 (dua) tipe dari prototype yaitu:

1. Evalutionary Prototype

Prototype yang secara terus menerus dikembangkan hingga prototype tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem.

1. Requirements Prototype

Prototype yang dibuat oleh pengembang dengan mendefiniskan fungsi dan prosedur sistem dimana pengguna atau pemilik sistem tidak bisa mendefinisikan sistem tersebut.

Kelebihan dari teknik pengembangan prototyping yaitu:

1. Menghemat waktu pengembangan
2. Menghemat biaya pengembangan
3. Pengguna atau pemilik sistem ikut terlibat dalam pengembangan, sehingga kemungkinan-kemungkinan terjadinya kesalahpahaman dalam sistem bisa diminimalisir.
4. Implementasi akan menjadi mudah, karena pengguna atau pemilik sistem sudah mempunyai gambaran tentang sistem.
5. Implementasi akan menjadi mudah, karena pengguna atau pemilik sistem sudah mempunyai gambaran tentang sistem.
6. Kualitas sistem yang dihasilkan baik.
7. Memungkinkan tim pengembangan sistem memprediksi dan memperkirakan pengembangan-pengembangan sistem selanjutnya.

Sedangkan kelemahannya adalah:

Pengguna atau pemilik sistem bisa terus menerus menambah kompleksitas sistem sehingga sistem menjadi sangat kompleks, hal ini bisa menyebabkan pengembang meninggalkan pekerjaannya sehingga sistem yang dikerjakan tidak akan pernah selesai.

1. **Rapid Application Development (RAD)**

Pengembangan sistem kompleks dengan waktu pengembangan sistem yang cepat, membuat para pengembang mencari solusi teknik pengembangan yang cepat tanpa mengurangi kualitas sistem yang dihasilkan. Dengan kondisi ini, dikembangkanlah Rapid Application Development (RAD).

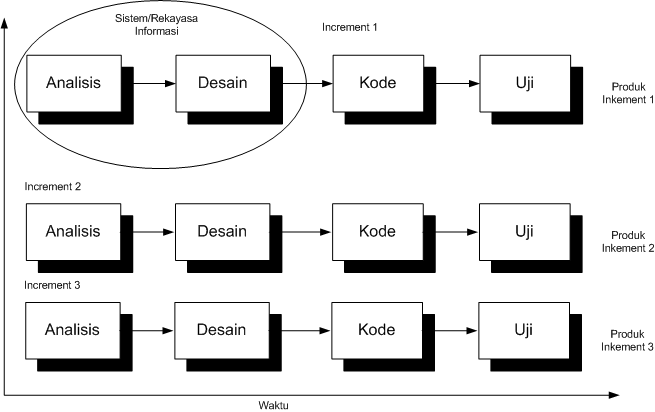
RAD merupakan memfokuskan pada kecepatan dalam pengembangan sistem untuk memenuhi kebutuhan pengguna dengan cangkupan yang lebih luas. Nama RAD dikenalkan oleh James Martin pada tahun 1991, yang mengacu pada life cycle pengembangan sistem. RAD mengadopsi teknik waterfall dan prototyping.

(Mc Leod & Schell, 2007) mengatakan ada 4 (empat) komponen RAD yaitu:

1. Manajemen, yaitu orang-orang (dari sisi user) yang berada pada level manajemen yang bisa beradaptasi dengan cepat untuk menggunakan metode baru.
2. Pengembang, yaitu tim pengembang sistem yang professional dalam menggunakan metode-metode pengembangan sistem dan tool yang dibutuhkan.
3. Metode, yaitu metode RAD yang dikenal dengan RAD Life Cycle.
4. Tools, yaitu Computer-Aided Software Engineering (CASE) dan Generation Language yang bisa memfasilitasi untuk pembuatan prototype dan pembuatan kode program. Sedangkan CASE tools lebih kepada dokumentasi dan perancangan database.
5. **Model Iteratif**

Mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe yang menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya. Model ini cocok untuk pengembang dengan turnover staf yang tinggi.

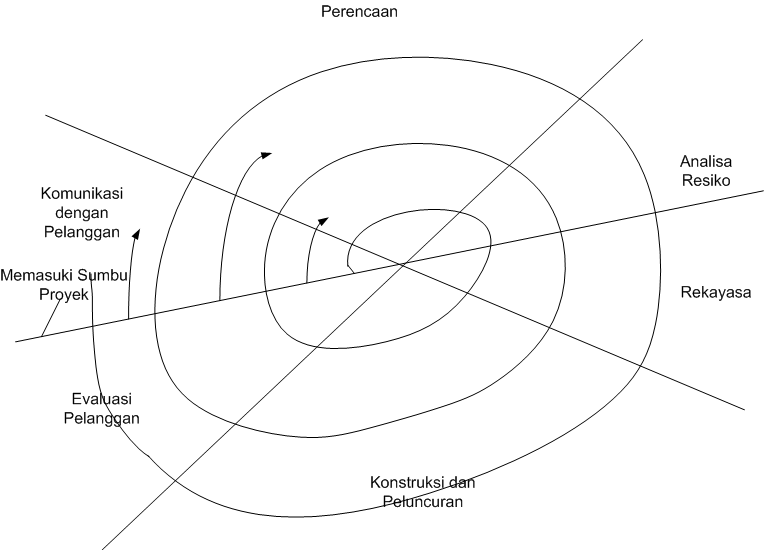
Model Inkremental dibuat untuk mengatasi kelemahan model waterfall yang tidak mengakomodasi iterasi dan mengatasi kelemahan dari metode prototipe yang memiliki proses terlalu pendek.

  
**Gambar 1.1. Model Iteratif**

1. **Model Spiral**

Memasangkan iteratif pada model prototipe dengan kontrol dan aspek sistematik yang diambil dari model air terjun yang menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya.

Model Spiral dibagi menjadi beberapa kerangka aktivitas atau disebut juga wilayah kerja (task region). Cocok digunakan untuk mengembangkan aplikasi dengan skala besar tetapi target waktu dan biaya tidak terlalu tinggi.



**Gambar 1.2. Model Spiral**

**Tahapan Model Spiral :**

1. Komunikasi dengan Pelanggan (*customer Communication*)

Untuk membangun komunikasi yang efektif antara pengembang (*developer*) dan pelanggan (*customer*)

1. Perencanaan (*Planning*)

Untuk mendefinisikan sumber daya, waktu, dan informasi yang terkait dengan proyek

1. Analisis Resiko (*Risk Analysis*)

Diperlukan untuk memperkirakan resiko dari segi teknis maupun manajemen.

1. Rekayasa

Diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak (dapat juga berupa prototipe).

1. Konstruksi dan Peluncuran (*Construction and release*)

Dibutuhkan untuk mengonstruksi, menguji, melakukan instalasi, dan menyediakan dukungan terhadap user.

1. Evaluasi Pelanggan (*customer Evaluation*)

Untuk mendapatkan umpan balikberdsarkan evaluasi representasi perangkat lunak yang dihasilkan dari proses rekayasa dan diimplementasikan pada tahap instalasi.

# BAB III ANALISA SISTEM INFORMASI

**Deskripsi :**

Mahasiswa mampu melakukan analisa system infomasi dengan mengumpulkan data yang dibutuhkan dalam membangun system informasi. Mahasiswa dapat menuliskan proses bisnis system berjalan dan menggambarkannya dengan Activity Diagram.

**Tujuan Pembelajaran:**

Mahasiswa mampu membuat:

1. Analisa system Informasi
2. Mengumpulkan data dengan metode PULTA
3. Merumuskan proses bisnis
4. Menggambar Activity Diagram

## 3.1. Analisa Kebutuhan Sistem

Pembangunan system informasi melibatkan metode, model, spesialis informasi dan user. Beberapa aplikasi dapat digunakan untuk permodelan sebuah system informasi seperti rational rose, IBM RSA, Microsoft Project, Power Designer, Mockup Builder. Untuk aplikasi pemodelan terdapat aplikasi yang mendukung peracangan basis data seperti DB2 Mysql, ODBC, Oracle yaitu CA ERwin Data Modeler. Sedang untuk merancang interface bisa kita gunakan aplikasi balsamiq mockup.

Dalam metode waterfall tahapan paling awal adalah perencanaan (planning) pada fase ini dicari tahu apakah dengan membangun system informasi dapat menyelesaikan permasalahan yang dihadapi perusahaan atau organisasi.

Fase yang kedua adalah analisa. Analisa adalah Suatu kegiatan yang dimulai dari proses awal didalam mempelajari serta mengevaluasi suatu bentuk permasalahan (case) yang ada.

Tujuan utama analisis sistem informasi yaitu:

1. Menentukan kelemahan dari proses-proses bisnis pada sistem lama untuk bisa menentukan kebutuhan dari sistem baru.
2. Menentukan tingkat kelayakan kebutuhan sistem baru.

Sedangkan Tipe tipe Kebutuhan sistem yaitu:

1. Operasional yang menjelaskan teknis bagaimana sistem baru akan beroperasi.
2. Kierja yang menjelaskan seberapa bagus kinerja perangkat lunak yang dikembangkan dalam mengolah data, menampilkan informasi, dan secara keseluruhan menyelesaikan proses bisnis.
3. Keamanan, berisi pernyataan tentang mekanisme pengamanan aplikasi, data, maupun transaksi yang akan diimplementasikan pada sistem.
4. Politik dan Budaya, isinya menyangkut atau berhubungan dengan isu politik dan budaya yang tidak menimbulkan persepsi negatif terhadap sistem.

## 3.2. Metode Pengumpulan Data

Pada fase analis kita berfokus pada pencarian informasi yang dianggap penting dalam menentukan kebutuhan system informasi. Adapun metode yang dapat dilakukan untuk mengumpulkan data/informasi adalah:

1.      Kuesioner

Kuesioner adalah daftar pertanyaan tertulis yang ditujukan kepada responden. Jawaban responden atas semua pertanyaan dalam kuesioner kemudian dicatat/direkam

2.      Observasi

Pengamatan melibatkan semua indera (penglihatan, pendengaran, penciuman, pembau, perasa). Pencatatan hasil dapat dilakukan dengan bantuan alat rekam elektronik

3.      Wawancara

Pengambilan data melalui wawancara /secara lisan langsung dengan sumber datanya, baik melalui tatap muka atau lewat telephone, teleconference. Jawaban responden direkam dan dirangkum sendiri oleh peneliti.

4.      Dokumen

Pengambilan data melalui dokumen tertulis mamupun elektronik dari lembaga/institusi. Dokumen diperlukan untuk mendukung kelengkapan data yang lain.

Adapun untuk sumber data/informasi kita bisa dapat dari dua sumber yaitu:

1.   Data Primer

Data penelitian yang diperoleh sendiri melalui: Wawancara, Observasi, Tes, Kuesioner (Daftar Pertanyaan), Pengukuran Fisik, Percobaan Laboratorium

2.   Data Sekunder

Data yang diperoleh dari sumber kedua, dokumentasi lembaga: Biro Pusat Statistik (BPS), Rumah sakit, Lembaga atau institusi

## 3.3. Perumusan Proses Bisnis

Bisnis Proses adalah suatu kumpulan pekerjaan yang saling terkait untuk  menyelesaikan suatu masalah tertentu. Suatu proses bisnis dapat dipecah  menjadi beberapa subproses yang masing-masing memiliki atribut sendiri tapi  juga berkontribusi untuk mencapai tujuan dari superprosesnya. Analisis  proses bisnis umumnya melibatkan pemetaan proses dan subproses di dalamnya hingga tingkatan aktivitas atau kegiatan**.**

Beberapa karakteristik umum yang dianggap harus dimiliki suatu proses bisnis adalah:

1. Definitif: Suatu proses bisnis harus memiliki batasan, masukan, serta keluaran yang jelas.
2. Urutan: Suatu proses bisnis harus terdiri dari aktivitas yang berurut sesuai waktu dan ruang.
3. Pelanggan: Suatu proses bisnis harus mempunyai penerima hasil proses.
4. Nilai tambah: Transformasi yang terjadi dalam proses harus memberikan nilai tambah pada penerima.
5. Keterkaitan: Suatu proses tidak dapat berdiri sendiri, melainkan harus terkait dalam suatu struktur organisasi.
6. Fungsi silang: Suatu proses umumnya, walaupun tidak harus, mencakup beberapa fungsi.

## 3.4. Penggambaran Activity Diagram Sistem Berjalan

Dalam buku ini akan disanpaikan contoh pembangunan system informasi perpustakaan, berikut adalah alur proses bisnis system berjalan yang dapat didefinisikan:

1. Pendaftaran Anggota.

Seorang calon anggota yang belum menjadi anggota perpustakaan dapat mendaftarkan diri dengan mengisi formulir pendaftaran. Setelah mengisi formulir tersebut, maka data calon anggota dapat di input kedalam sistem oleh pustakawan. Pustakawan mencetak kartu anggota dan menyerahkan ke anggota perpustakaan.

2. Pengolahan

Pustakawan harus memasukkan data koleksi bahan pustaka yang dibelinya ke dalam sistem. Proses pengelolan akan dilaksanakan dan pustakawan dapat mencetak label koleksi untuk masing-masing koleksi bahan pustaka.

3. Peminjaman.

Koleksi bahan pustaka yang telah melalui proses pengolahan, dapat dipinjamkan kepada anggota. Saat seorang anggota meminjam koleksi bahan pustaka, pustakawan harus memasukkan nomor keanggotaan, dan sistem akan menyimpan data peminjaman serta menampilkan form untuk menambahkan data lengkap peminjaman. Pustakawan mengisikan nomor dari koleksi bahan pustaka yang akan dipinjam dan tanggal pengembalian.

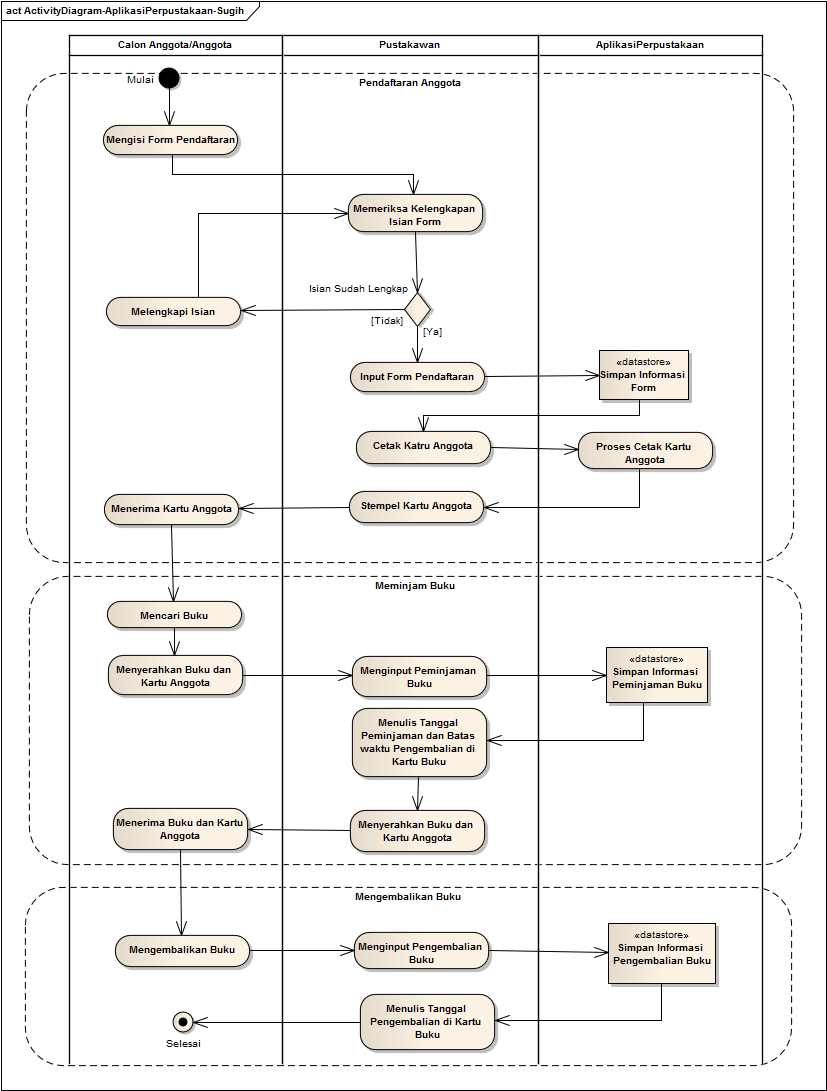
4. Pengembalian.

Pustakawan harus mencatat data pengembalian, pada saat anggota mengembalikan buku pinjamannya. Pustakawan mengisikan nomor dari koleksi bahan pustaka yang dikembalikan pada form Pengembalian. Pada form ini terdapat informasi seperti anggota peminjam, koleksi bahan pustaka yang telah dipinjam tapi belum dikembalikan, tanggal pengembalian, denda yang harus dibayar, dan perpanjangan waktu peminjaman.

Penggambaran proses bisnis dilakukan menggunakan activity diagram. Activity diagram adalah menggambarkan  proses bisnis dan urutan aktivitas dalam sebuah proses. Memperlihatkan urutan aktifitas proses pada system.

Tujuan utama dari Activity Diagram adalah menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses agar lebih mudah dipahami. Activity diagram juga digunakan untuk menunjukkan aliran pesan dari satu aktivitas ke aktivitas lainnya. Adapun tujuan pembuatan Activity Diagram lainnya yaitu :

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
2. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis
3. Membantu memahami proses secara keseluruhan
4. Menggambarkan aliran paralel, bercabang dan bersamaan dari sistem



**Gambar 3.1. Activity Diagram Sistem Berjalan**

# BAB IV DESAIN DATABASE & STRUKTUR KODE

**Deskripsi :**

Mahasiswa dapat merancang desain database dari suatu dokumen, membuat spec file dan membuat struktur pengkodean yang baik.

**Tujuan Pembelajaran:**

Mahasiswa mampu membuat:

1. Entity Relationship Diagram dengan menggunakan symbol ERD dan membuat Kardinality
2. Spesifikasi File dari database yang ada
3. Pengkodean setiap primary key dari tabel dalam database

## 4.1. Pengertian ERD

Ada sejumlah cara dalam mempresentasikan model data menurut (Fathansyah, 2018), secara umum dikelompokkan kedalam dua kelompok:

1. Model logic data berdasarkan OBJEK (*Object Based Logical Model*) yang terdiri dari:
2. Model Keterhubungan Entitas (*Entity Relationship Model*)
3. Model Berorientasi Objek (*Object Oriented Model*)
4. Model Data Semantik (*Semantic Data Model*)
5. Model Data Fungsional (*Functional Data Model*)
6. Model logic data berdasarkan Record (*Record Based logical Model*) yang terdiri dari:
7. Model Relasional (*Relational Model*)
8. Model Hierarkis (*Hierarchial Model*)
9. Model Jaringan (*Network Model*)

Proses pembuatan database dimulai dengan membuat desain database, ERD merupakan salah satu model data yang dapat menggambarkan desain database dengan mudah. Menurut (Mulyani, 2016) ERD merupakan tools yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antar entitas (*relationship*) secara abstrak (konseptual). Ada tiga fungsi utama ERD yaitu:

1. Sebagai alat untuk memodelkan hasil dari analisa data
2. Sebagai alat untuk memodelkan data konseptual (lojikal)
3. Sebagai alat untuk memodelkan objek-objek dalam suatu system (Dasar dari *Object Diagram/Class Diagram*)

## 4.2. Simbol pada ERD

Dalam pembuatan ERD ada berbagai macam symbol yang digunakan tetapi pada intinya ada 3 simbol utama yaitu:

1. Entity
2. Atribut
3. Relationship
4. **Entity**

Simbol Entity adalah sebuah objek yang keberadaannya dapat dibedakan terhadap objek lain, Entitas dapat berupa orang, benda, tempat, kejadian, konsep. Entity digambarkan dengan bentuk persegi panjang. Contoh dari entity katagori orang: MAHASISWA, DOSEN, PEMASOK,PENJUAL, kategori Benda: MOBIL, MESIN, RUANGAN, kategori Tempat: NEGARA, DESA, KAMPUNG, kategori kejadian: PENJUALAN, REGISTRASI Kategori Konsep: REKENING, KURSUS.

Entity dapat dibedakan berdasarkan sifarnya menjadi dua yaitu strong entity dan week entity. Strong entity yaitu entity set yang satu atau lebih atributnya digunakan oleh entity set lain sebagai key. Digambarkan dengan empat persegi panjang. Misal: E adalah sebuah entity set dengan atribute-atribute a1, a2,..,an, maka entity set tersebut direpresentasikan dalam bentuk tabel E yang terdiri dari n kolom, dimana setiap kolom berkaitan dengan atribute-atributenya. Weak Entity set, Entity set yang bergantung terhadap strong entity set. Digambarkan dengan empat persegi panjang bertumpuk. Misal:

A adalah weak entity set dari atribute-atribute a1, a2, .., ar dan B adalah strong entity set dengan atribute-atribute b1, b2,..,bs, dimana b1 adalah atribute primary key, maka weak entity set direpresentasikan berupa table A, dengan atribute-atribute {b1} u {a1,a2,.., ar}

1. **Atribut**

Simbol Atribute adalah karakteristik dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tsb. Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

**Jenis-jenis atribut :**

Atribut Simple

Atribut yang bernilai tunggal. Di simbolkan dengan elips.

Atribut Key

Atribut yang digunakan untuk menentukan suatu entity secara unik. Di simbolkan dengan elips dan diberi garis bawah pada nama atribut.

Atribut Multivalue

Atribut yang memiliki sekelompok nilai untuk setiap instan entity. Di simbolkan dengan elips didalam elips

Atribut Composite

Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu contohnya adalah atribut nama pegawai yang terdiri dari nama depan, nama tengah dan nama belakang. Di simbolkan dengan sebuah elips yang memiliki cabang sesuai dengan pecahan atributnya.

Atribut Derivarif

Suatu atribut yg dihasilkan dari atribut yang lain. Sehingga umur yang merupakan hasil kalkulasi antara Tgl Lahir dan tanggal hari ini. Sehingga keberadaan atribut umur bergantung pada keberadaan atribut Tgl Lahir. Di simbolkan dengan elips dengan garis putus-putus.

PEGAWAI

Contoh 4.2:

**Gambar 4.1. Contoh Jenis Atribut**

**C. Relationship**

Relationship adalah hubungan yang terjadi antara satu atau lebih entity. Relationship set adalah kumpulan relationship yang sejenis. Relasi memiliki tiga derajat yaitu:

1. Derajat satu (*Unary Degree*) maksudnya adalah entitas memiliki hubungan dengan dirinya sendiri. Contoh: seorang pegawai menyampaikan laporan ke atasannya yang juga pegawai.

**PEGAWAI**

**LAPOR**

**Gambar 4.2 *Unary Degree***

1. Derajat dua (*Binary Degree*) maksudnya adalah satu entitas akan memiliki hubungan dengan entitas lain. Contoh: Pegawai bekerja pada satu departemen.

**PEGAWAI**

**PEGAWAI**

**PEGAWAI**

**Gambar 4.3. *Binary Degree***

1. Derajat tiga (*Ternary Degree*) maksudnya adalah tiga entitas yang saling berhubungan dalam satu relasi. Contoh: Pegawai bekerja pada suatu departemen pada satu kota.

**PEGAWAI**

**KERJA**

**PROYEK**

**DEPARTEMEN**

**Gambar 4.4. *Ternary Degree***

Selain itu relasi juga memiliki tingkat hubungan kardinalitas yang dimaksud dengan kardinality adalah Banyaknya entity yang bersesuaian dengan entity yang lain melalui relationship. Ada tiga macam kardinality:

1. *One to One Relationship*, yaitu setiap satu objek entitas hanya terhubung dengan satu objek entitas lain. Contoh: setiap satu orang pegawai memiliki satu kendaraan.

**PEGAWAI**

**MILIK**

**KENDARAAN**

**1**

**1**

**Gambar 4.5. *One To One Relationship***

1. *One to Many Relationship*, yaitu setiap satu objek entitas bisa memiliki hubungan lebih dari satu dari entitas lain. Contoh: Satu Dosen dapat mengkoordinir beberapa matakuliah, tetapi satu matakuliah hanya di koordinir oleh satu dosen.

**PEGAWAI**

**KERJA**

**PEGAWAI**

**1**

**N**

**Gambar 4.6.  *One To Many Relationship***

1. *Many to Many Relationship*, yaitu banyak objek dalam satu entitas dapat memiliki hubungan dengan banyak objek dari entitas lain. Contoh: satu pegawai dapat bekerja pada beberapa proyek dan satu proyek bisa dikerjakan oleh beberapa pegawai.

**PEGAWAI**

**KERJA**

**PROYEK**

**M**

**N**

**Gambar 4.7. *Many To Many Relationship***

## 4.3. Tahapan Pembuatan & Contoh ERD

Dalam pembuatan ERD dapat dilakukan melalui lima tahapan berikut ini adalah tahapannya:

1. Tahap pertama, tentukan semua entitas yang terlibat.
2. Tahap kedua, tentukan atribut dari setiap entitas
3. Tahap ketiga, Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta *foreign key* nya
4. Tahap keempat, Tentukan derajat relasi (*cardinality*) dengan matriks relasi untuk setiap himpunan relasi
5. Tahap kelima, Lengkapi Entitas dengan atribut non kunci

Berikut ini akan kami contohkan untuk pembuatan ERD dengan studi kasus perpustakaan.

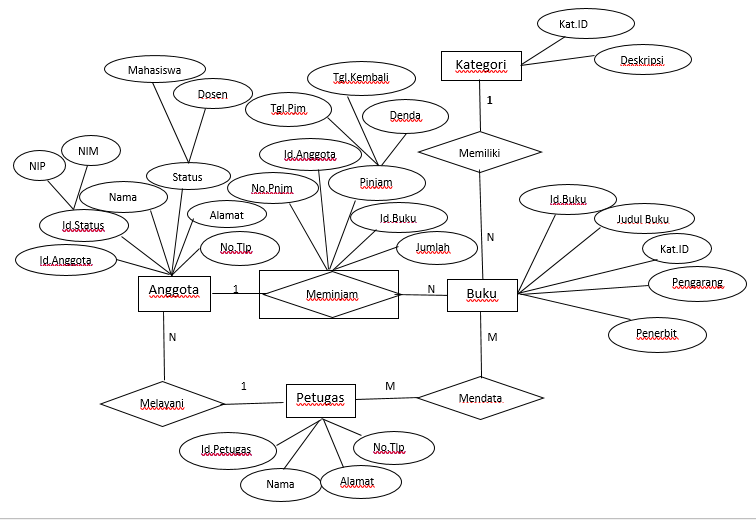
**Tahap pertama**: entitas yang terlibat pada perpustakaan adalah: Anggota, Petugas, Buku dan Kategori

**Tahap Kedua**: Dari entitas Anggota terdapat atribut kunci IdAnggota, Dari entitas Petugas terdapat atribut kunci IdPetugas. Dari entitas Buku terdapat atribut kunci IdBuku dan foreignkey KatId. Dari entitas Kategori terdapat atribut kunci KatID.

**Tahap Ketiga**: entitas anggota berelasi dengan buku dengan relasi meminjam, entitas buku berelasi dengan kategori. Dan relasi meminjam memiliki primary key nopinjam dan memiliki foreign key IdAnggota dan IdBuku.

**Tahap Keempat**: cardinality entitas anggota dengan buku adalah one to many dimana satu orang anggota bisa meminjam lebih dari satu buku, tetapi satu buku hanya bisa dipinjam oleh satu anggota, cardinality entitas buku dengan kategori many to one dimana satu satu kategori bisa untuk beberapa buku tetapi satu buku hanya satu kategori.

**Tahap Kelima**: Dari entitas Anggota terdapat atribut (IdAnggota, Idstatus yang terdiri dari NIP dan NIM, Nama, Status yang terdiri dari dosen dan mahasiswa, alamat, notlp). Dari entitas Petugas terdapat atribut (IdPetugas, Nama, Alamat, NoTlp). Dari entitas Buku terdapat atribut (IdBuku, Judul, Pengarang, Penerbit, IdKategori). Dari entitas Kategori terdapat atribut (KatID, Deskripsi). Pada relasi meminjam terdapat foreign key Id Anggota dan IdBuku dan terdapat atribut tambahan atribut pinjam (tglpinjam, tglkembali, denda), dan jumlah. Dengan adanya atribut tambahan tersebut dapat diketahui data peminjaman, sehingga dari tahapan tersebut maka diperoleh gambaran ERD sebagai berikut:



**Gambar 4.8. Entity Relationship Diagram**

## 4.4. Spesifikasi File

Menjelaskan tentang file atau tabel yang terbentuk dari transformasi ERD (dan atau file-file penunjang untuk web). File-file ini tersimpan pada <<namaDatabase>> dengan parameter-parameter yang telah ditentukan. Spesifikasi file mirip dengan meta data pada database, yang berisi informasi dari tiap tabel yang ada pada database, berikut adalah contoh dari spesifikasi file:

Nama *File* : Buku

Akronim : buku.MYD

Tipe File : File Master

*Acces File* : *Random*

Panjang *Record* : 250 *bytes*

*Field Key* : kd\_buku

*Software*  : MySQL

Nama file adalah sebutan dari tabel dalam database sedangkan akronim adalah nama tabel dalam database disertai dengan extention filenya, untuk tipe file dipilih sesuai dengan tipe file dalam computer yaitu:

1. **File Master**

Jenis informasi cenderung tetap, tetapi isi informasinya dapat sering berubah, dan selalu diperbarui menggunakan file transaksi.

**1. File Referensi**

Data yang tetap dimana pengolahan terhadap data tersebut memerlukan waktu yang lama. Data yang terdapat pada file referensi ini sebagai contoh adalah file Mahasiswa, yang apabila akan diperbaiki (diedit) untuk jangka waktu yang lama, misalnya jika terjadi perbaikan pada pengisian data untuk alamat (jika mahasiswa tersebut pindah alamat rumah) maka isi data pada alamat untuk mahasiswa yang bersangkutan harus diperbaiki.   
  
**2. File Dinamik**

Data yang ada dalam file berubah tergantung transaksi,atau berubah secara berkala sebagai hasil dari suatu transaksi. Misalkan saja file mata kuliah yang di dalamnya terdapat informasi tentang seluruh mata kuliah yang terdapat disuatu lembaga pendidikan. Isi dari file mata kuliah dapat diperbaiki (di-edit) jika kondisi yang ada di masa depan mengharuskan adanya pergantian mata kuliah dengan nama mata kuliah yang baru dan sks yang juga baru, maka perbaikan data untuk isi file mata kuliah mutlak dilakukan. Hal ini yang disebut dengan penyetaraan mata kuliah.

**B. File Input (Transaksi)**

Digunakan untuk memperbarui/meng-update file master dengan informasi yang baru  Meng-update dapat berupa penambahan record atau penghapusan record .Berisi data masukan yang berupa data transaksi dimana data-data tersebut akan diolah oleh komputer . File transaksi akan senantiasa mengalami perubahan sesuai dengan periode waktu tertentu . Macam file transaksi yang ada pada lembaga pendidikan adalah file ujian, file nilai dan file pembayaran kuliah.

**C. File Laporan (File Output)**

File ini berisi informasi yang akan ditampilkan dalam sebuah laporan.  Isi dari Laporan yang dihasilkan biasanya berasal dari penggabungan file master dan file transaksi, tetapi tidak semua isi dari file-file tersebut ditampilkan, melainkan hanya informasi tertentu saja yang ditampilkan sesuai dengan laporan yang diinginkan. File ini dapat dicetak di kertas, atau hanya ditampilkan saja di layar monitor.

**D. File Pelindung(Backup File)**

Merupakan salinan (copy) data-data yang masih aktif di database (current database) pada suatu waktu tertentu dapat berupa file master maupun file transaksi. Adapun isi (informasi) dari file backup ini sama persis dengan file aslinya. Jika ada yang diperbaiki maka hasil perbaikan data tersebut harus dibuatkan kembali backup dari file yang diperbaiki. Frekuensi penggunaannya tergantung pada frekuensi perubahan isi file dan banyaknya rekaman transaksi yang digunakan untuk pembaharuan file. Digunakan sebagai pelindung atau cadangan apabila file master mengalami kerusakan atau hilang.   
  
**E. File Kerja(Work File)**

File ini berisi data hasil pemrosesan yang sifatnya sementara (temporary) dalam arti hanya numpang lewat saja. Datanya berupa record yg disusun sedemikian rupa sehingga dapat dipakai oleh program lain sebagai input.Biasanya file ini dibuat pada waktu proses sortir.

**F. File Pustaka(Library File)**

File yang berisi program aplikasi, program utilitas dan program bantu yang dapat berfungsi untuk mempercepat dan optimalisasi dari pengolahan data. Contohnya : file gaji yang mempunyai keterhubungan dengan file pendidikan pegawai (untuk mencari jenjang pendidikan dari seorang karyawan),file absensi(untuk mengetahui data kehadiran pegawai) dan file lembur(untuk mengetahui intensitas kelebihan jam kerja yang dimiliki oleh seorang pegawai).

**G. File History (Riwayat Hidup)**

File yang berisi data di masa lalu yang dipakai sebagai bahan pengecekan dan pencocokkan kevalidasian data di masa mendatang. File ini biasanya digunakan untuk periode waktu tertentu(lama), sehingga beberapa informasi harus dikumpulkan pada file riwayat hidup yang akan jarang diproses. Contoh : informasi tentang pasien yang keluar tidak perlu disimpan pada file yang sekarang ada.Record hanya dipindahkan pada file history sehingga tinggal berisi file pasien yang sedang dirawat atau dijadwalkan akan kembali.Data pasien yang keluar disimpan di file history yang berguna untuk menyusun laporan di kemudian hari.

Sedangkan untuk akses file adalah bagaimana cara kita mengakses data pada media penyimpanan. Ada beberapa teknik akses file yaitu:

**1.Sequential File.**

Merupakan suatu cara/metode penyimpanan dan pembacaan data yang dilakukan secara berurutan.   
Metode ini mempunyai ciri-ciri sebagai berikut:

a.Data akan disimpan sesuai dengan urutan masuknya.

b.Pencarian rekaman tertentu dilakukan record demi record.

c.Pembacaan data, juga dilakukan secara berurutan.

Metode ini baik untuk digunakan apabila pengolahan terhadap basis data bersifat periodic dan menyeluruh.

Penambahan Record , Penambahan record ke dalam suatu file, maka posisi record terakhir harus diketahui.Karena record baru diletakkan pada posisi setelah record terakhir.  Penambahan record baru tersebut diletakkan pada posisi akhir dari file.  Untuk penyisipan record, posisi yang akan ditempati oleh record baru harus diketahui dengan jelas.Posisi ini dapat diketahui dengan cara memeriksa salah satu isi field dari   
suatu record.

Perubahan Record, Perubahan record merupakan modifikasi terhadap isi record dari satu  

**2. Random.**

Merupakan suatu cara/metode penyimpanan dan pembacaan data yang dilakukan secara acak atau langsung. Dalam hal ini, tempat penyimpanan data sudah diatur sedemikian rupa, sehingga setiap data akan tersimpan didalam tempat-tempat yang telah ditentukan sesuai dengan nomor data yang dimiliki-nya.  Pemrosesan memerlukan file record, yg ditampilkan menjadi 2 bagian yaitu field kunci dan record data. Untuk menghubungkan nilai kunci dengan alamat, digunakan direktori yang terdiri dari 2 kolom, yaitu 1 untuk nilai kunci, dan 1 untuk alamat Teknik untuk menentukan alamat data yang disimpan dengan akses langsung, yaitu:

a. Pengalamatan absolut, yaitu alamat aktual dari record yang diinginkan (nilai key = alamat absolut).

b. Pengalamatan relatif,(nilai key = alamat relatif)alamat relatif yaitu urutan record dalam berkas

c. Pengalamatan indeks, yaitu menentukan lokasi dengan satu nilai kunci   
  
Dalam metode ini kunci rekaman ditransformasikan ke alamat penyimpanan dalam media fisik secara acak(random).Metode ini akan menimbulkan beberapa masalah,yaitu adanya alamat yang muncul lebih dari satu kali (K1 dan K2),dan ada alamat yang tidak pernah muncul sama sekali. Permasalahan seperti ini diatasi dengan teknik overflow location, yaitu dengan menggunakan alamat yang ada disampingnya, dg cara K1 disimpan dalam home addressnya, sedangkan K2 disimpan dalam open addressing atau separate overflow.

**3. Indexed Sequential.**

Metode ini mempunyai ciri-ciri sebagai berikut:

a.Gabungan antara metode sequential dan random.

b.Record disimpan secara berurutan dengan menggunakan kunci (indeks urutan).

c.Masing-masing record diberi indeks.

d.Pengalamatan dilakukan secara acak.

e.Perlu penyimpanan tambahan, yaitu untuk file indeks

f.Index ini diakhiri denga adanya suatu pointer(penunjuk)yang bisa menunjukkan secara jelas posisi data yang selengkapnya.

g.Index yang ada juga merupakan record-key (kunci record), sehingga kalau record key ini dipanggil,maka seluruh data juga akan ikut terpanggil.

Untuk panjang record kita hanya perlu menjumlahkan panjang dari setiap field yang ada di tabel. Sedangkan untuk field key kita cantumkan primary key dari tabel yang dimaksud, untuk softaware adalah software yang digunakan dalam membuat database.

## 4.5. Struktur Kode

Cara untuk menyatakan data dalam bentuk lain agar efisien dalam ruang penyimpanan adalah dengan melalui pengkodean (data coding), dari pemakaiannya kita bisa membedakan pengkodean internal (*system coding*) dan eksternal (*user-defined coding*). Pengkodean eksternal mewakili pengkodean yang telah dilakukan secara terbuka dan dikenal oleh pemakai awam (*end-user*) begitu sebaliknya.

Menurut fatansyah 2018, ada tiga bentuk pengkodean yang dapat kita pilih, yaitu:

1. Sekuensial, dimana pengkodean dilakukan dengan mengasosiasikan data dengan kode terurut. Contoh: nilai mutu (“Sempurna”, “ Baik”, “Cukup”,”Kurang”) di kodekan dengan (“A”, “B”, “C”, “D”)
2. Mnemonic, dimana pengkodean dilakukan dengan membentuk suatu singkatan dari data yang ingin di kodekan. Contoh: Laki-laki dan perempuan yang dikodekan dengan L dan P.
3. Blok, dimana pengkodean dinyatakan dalam format tertentu. Contoh: Nim yang terdiri dari 8 digit dimana dua digit pertama mewakili kode jurusan, dua digit selanjutnya mewakili tahun masuk dan empat digit terakhir mewakili nomor urut pendaftaran.

# BAB V *UNIFIED MODELING LANGUAGE*

**Deskripsi:**

Membahas cara pengimplementasian *Unified Modeling Language* (UML) dalam merancang suatu program ataupun sistem informasi, khususnya dengan *use case*, *activity*, *class*, dan *sequence* *diagram*.

**Tujuan Pembelajaran:**

Setelah melakukan bagian ini mahasiswa mampu:

* + - 1. Menggambarkan interaksi antara pengguna dengan sistem dalam *use case diagram*.
      2. Memodelkan proses bisnis suatu sistem informasi dalam *activity diagram*.
      3. Menggambarkan class dan relasinya dengan class lain dalam *class diagram*.
      4. Mengilustrasikan pesan yang lewat di antara objek untuk use case tertentu dalam *sequence diagram*.

## PENGERTIAN *UNIFIED MODELING LANGUAGE* (UML)

*Unified Modeling Language* (UML) adalah suatu teknik untuk memodelkan sistem. UML ditemukan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh. UML versi terbaru, yaitu versi 2.5, terdiri dari lima belas diagram. Diagram-diagram tersebut dibagi ke dalam dua kelompok, yaitu *structure diagram* dan *behavior diagram*. *Structure diagram* menggambarkan data dan hubungan statis dalam suatu sistem informasi. *Structure diagram* terdiri dari *package*, *object, component, class, deployment, composite structure,* dan *profile diagram.* *Behavior diagram* menggambarkan hubungan dinamis diantara objek yang mewakili sistem informasi bisnis. *Behavior diagram* terdiri dari *sequence, timing, interaction overview, activity, use case, protocol state machine*, *communication* dan *behavior state machine diagram* (Dennis, Wixom, & Tegarden, 2015).

### USE CASE DIAGRAM

*Use case* digunakan untuk menggambarkan fungsi dasar dari sebuah sistem informasi. *Use case* mendeskripsikan cara sistem bisnis berinteraksi dengan lingkungannya (Dennis et al., 2015). Berikut adalah elemen-elemen dari *use case diagram:*

1. *Actor/Role*



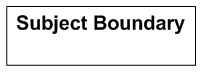
*Actor/*role adalah orang atau sistem lain yang berinteraksi dengan sistem saat ini.

1. *Use Case*



*Use case* adalah bagian utama dari fungsionalitas sistem. Bisa *extend* (memperluas) *use case* lainnya. Ditempatkan di dalam *system boundary* (batasan sistem). Dilabeli dengan kata kerja – frase kata benda.

1. *Subject Boundary*



Berisi nama dari sistem yang diletakkan di dalam atau di bagian atas *boundary*. Mewakili ruang lingkup sistem. *Actor* berada di luar ruang lingkup sistem.

1. *Association Relationship*



Menghubungkan *actor* dengan *use case*. Menunjukkan komunikasi dua arah (Menunjukkan komunikasi satu arah jika menggunakan tanda panah). Tanda \* untuk keragaman dari asosiasi (*multiplicity of the association*). Namun umumnya hanya digambarkan garis saja.

1. *Include Relationship*



Memasukkan satu *use case* dalam *use case* lainnya. Perilaku (*behavior*) yang harus terpenuhi agar sebuat *event* dapat terjadi, di mana kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya. Tanda panah mengarah dari *base use case* (pusat) menuju ke *use case* yang di-include.

1. *Extend Relationship*

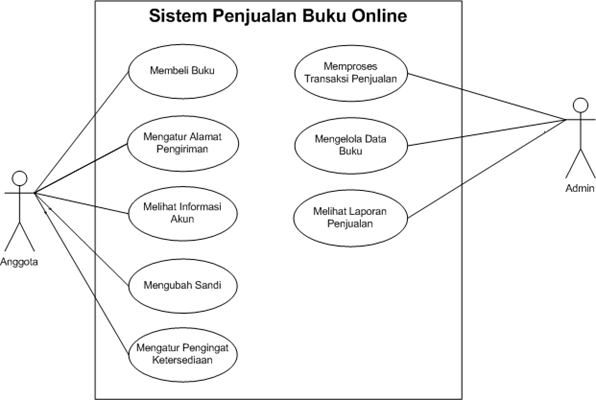


Memperluas *use case* untuk memasukkan perilaku opsional. Tanda panah mengarah dari *use case* tambahan ke *base use case* (pusat).

1. *Generalization Relationship*

Mewakili use case khusus untuk use case yang lebih umum. Tanda panah mengarah dari use case khusus (specialized) ke use case yang lebih umum.

Contoh dari *use case diagram* bisa dilihat pada gambar 5.1.



**Gambar 5.1. Contoh *Use Case Diagram* Untuk Sistem Pendaftaran Siswa**

### ACTIVITY DIAGRAM

*Activity diagram* mengilustrasikan kegiatan utama dan hubungan di antara kegiatan dalam suatu proses (Dennis et al., 2015). Berikut adalah elemen-elemen dari *activity diagram*:

1. *Action*



Perilaku yang sederhana dan tidak dapat diuraikan. Dilabeli dengan namanya.

1. *Activity*



Digunakan untuk mewakili sekumpulan tindakan (action). Dilabeli dengan namanya.

1. *Object Node*



Digunakan untuk mewakili sebuah objek yang terhubung ke sekumpulan arus objek. Dilabeli dengan nama classnya.

1. *Control Flow*



Menunjukkan urutan eksekusi.

1. *Object Flow*



Menunjukkan aliran suatu objek dari satu aktivitas (atau tindakan) ke aktivitas (atau tindakan) lain.

1. *Initial Node*



Menggambarkan awal dari serangkaian tindakan atau kegiatan.

1. *Final-activity Node*



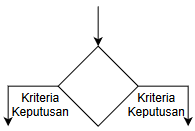
Digunakan untuk menghentikan semua arus kontrol dan arus objek dalam suatu aktivitas (atau tindakan).

1. *Final-flow Node*



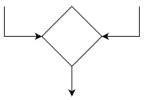
Digunakan untuk menghentikan aliran kontrol atau aliran objek tertentu.

1. *Decision Node*



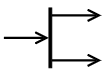
Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur. Dilabeli dengan kriteria keputusan untuk melanjutkan ke jalur tertentu.

1. *Merge Node*



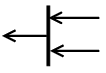
Digunakan untuk menyatukan kembali berbagai jalur keputusan yang dibuat menggunakan simpul keputusan.

1. *Fork Node*



Digunakan untuk membagi perilaku menjadi seperangkat aktivitas yang paralel atau bersamaan dari aktivitas (atau tindakan).

1. *Join Node*



Digunakan untuk menyatukan kembali serangkaian arus aktivitas (atau tindakan) yang paralel atau bersamaan.

1. *Swimlane*



Digunakan untuk memecah diagram aktivitas menjadi baris dan kolom untuk menetapkan kegiatan individu (atau tindakan) kepada individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas (atau tindakan). Dilabeli dengan nama individu atau objek yang bertanggung jawab.

Contoh dari *activity diagram* bisa dilihat pada gambar 5.2.



**Gambar 5.2. Contoh *Activity Diagram* Untuk Sistem Pendaftaran Siswa**

### CLASS DIAGRAM

*Class diagram* adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan kelas, yang meliputi perilaku dan keadaan, dengan hubungan antar kelas (Dennis et al., 2015). Berikut adalah elemen-elemen dari *class diagram*:

*Class*



Mewakili orang, tempat, atau hal-hal yang dibutuhkan sistem untuk menangkap dan menyimpan informasi. Memiliki nama yang diketik dengan huruf tebal dan berpusat di bagian atas kotak. Memiliki daftar atribut di kotak tengahnya. Memiliki daftar operasi di kotak bawahnya. Tidak secara eksplisit menunjukkan operasi yang tersedia untuk semua kelas.

*Attribute*



Merupakan properti yang menggambarkan keadaan suatu objek. Dapat diturunkan dari atribut lain, ditampilkan dengan menempatkan garis miring sebelum nama atribut.

*Operation*

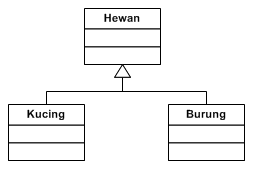


Mewakili tindakan atau fungsi yang dapat dilakukan oleh kelas. Dapat diklasifikasikan sebagai konstruktor, permintaan, atau operasi pembaruan. Termasuk tanda kurung yang mungkin berisi parameter atau informasi yang diperlukan untuk melakukan operasi.

*Generalization*

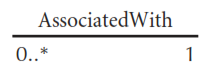


Mewakili hubungan sejenis di antara beberapa kelas. Contoh dari *generalization* bisa dilihat pada gambar 5.3.



**Gambar 5.3. Contoh *Generalization***

*Association*



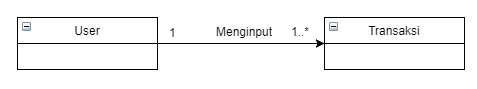
Mewakili hubungan antara beberapa kelas atau kelas dan dirinya sendiri. Dilabeli menggunakan frasa kata kerja atau nama peran, yang lebih baik mewakili hubungan. Dapat ada di antara satu atau beberapa kelas. Berisi simbol multiplisitas, yang mewakili waktu minimum dan maksimum instance kelas dapat dikaitkan dengan instance kelas terkait.

Tujuan utama dari class diagram adalah untuk menunjukkan relasi, atau asosiasi, yang class miliki dengan class lainnya. Relasi juga memliki **multiplicity**, yang mendokumentasikan bagaimana class bisa diasosiasikan dengan class lainnya. Format **multiplicity**: angka minimal..angka maksimal. Penjelasan lebih detail mengenai multiplicity bisa dilihat pada tabel 5.1.

**Tabel 5.1. Multiplicity**

|  |  |  |  |
| --- | --- | --- | --- |
| Hanya satu | 1 |  | Dalam satu departemen hanya memiliki satu bos |
| Nol atau lebih | 0..\* |  | Seorang karyawan bisa saja tidak memiliki anak (0) atau memiliki banyak anak (\*) |
| Satu atau lebih | 1..\* |  | Satu bos bertanggungjawab untuk satu atau lebih karyawan |
| Nol atau satu | 0..1 |  | Seorang karyawan bisa saja belum/tidak menikah (memiliki 0 pasangan) atau menikah (memiliki satu pasangan) |
| Range yang spesifik | 2..4 |  | Seorang karyawan bisa mengambil libur (cuti) mulai dari 2 sampai 4 hari setiap tahunnya |
| Multiple | 1..3,5 |  | Seorang karyawan adalah anggota dari satu sampai 3 atau 5 komite |

Untuk menentukan kadinalitas (derajat relasi) antar dua entitas/class, **cukup ambil angka maksimal dari tiap multiplicity**. Penjelasannya bisa dilihat pada gambar 5.4.



1..M

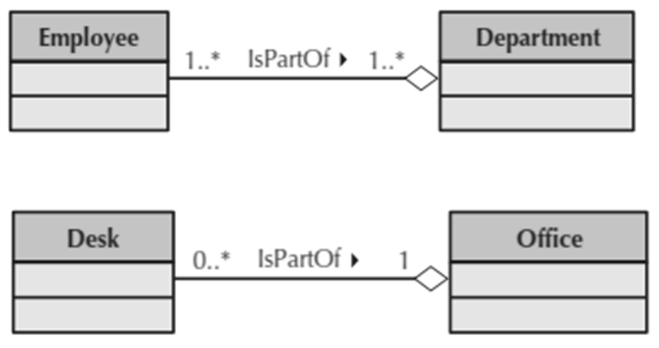
**Gambar 5.4. Kardinalitas**

Kardinalitas dari gambar di atas adalah 1..M. Membaca dari kiri ke kanan: menunjukkan 1 user menginput minimal 1, maksimal banyak transaksi. Membaca dari kanan ke kiri: 1 Transaksi hanya bisa diinput oleh 1 orang. Jika ada relasi 1..M (di sebelah kiri) dan 1..M (di sebelah kanan), maka kardinalitasnya **M..N (many to many)**.

*Aggregation*

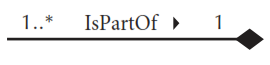


Merupakan hubungan sebagian (a part of) yang logis diantara beberapa kelas atau suatu kelas dengan kelas itu sendiri. Aggregation terbuat dari objek-objek yang bisa dibagi atau ditukar. Angka di sebelah kanan boleh 1, boleh juga banyak (\*). Contoh dari *aggregation* bisa dilihat pada gambar 5.5.

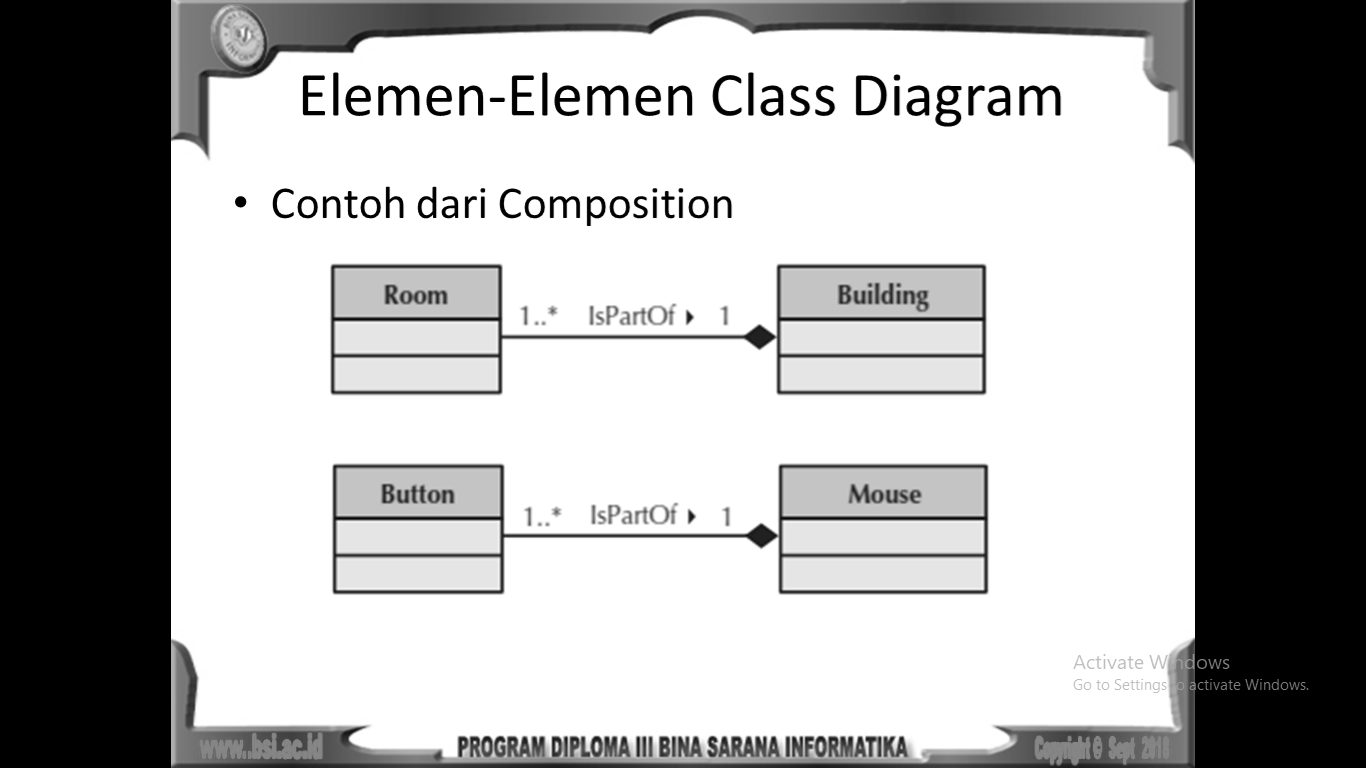


**Gambar 5.5. Contoh *Aggregation***

*Composition*



Mewakili hubungan fisik antara beberapa kelas atau kelas dan kelas itu sendiri. *Composition* terdiri dari objek-objek yang tidak bisa dibagi atau ditukar dan hidup selama *composite* objek hidup. Angka di sebelah kanan hanya boleh 1. Contoh dari *composition* bisa dilihat pada gambar 5.6.



**Gambar 5.6. Contoh *Composition***

Contoh *class diagram* bisa dilihat pada gambar 5.7.



**Gambar 5.7. *Class diagram***

**Penjelasan diagram:**

1. User boleh menginput satu atau banyak data barang. Sedangkan satu data barang diinput oleh satu orang user saja.
2. User boleh menginput satu kali atau sebanyak mungkin transaksi penjualan. Sedangkan dalam satu transaksi boleh diinput oleh satu orang user saja.
3. Detail transaksi merupakan Agregasi Komposit, dalam artian Transaksi memiliki Detail transaksi atau Detail transaksi tidak akan ada kalau kelas Transaksi tidak ada.
4. Dalam satu detail transaksi harus tercatat kode barang minimal satu atau banyak.

### SEQUENCE DIAGRAM

Sequence diagram menunjukkan pesan yang lewat di antara objek untuk *use case* tertentu dari waktu ke waktu. *Sequence* *diagram* mengilustrasikan objek-objek yang berpartisipasi di dalam suatu *use case* (Dennis et al., 2015). Berikut adalah elemen-elemen dari *sequence diagram*:

1. *Actor*

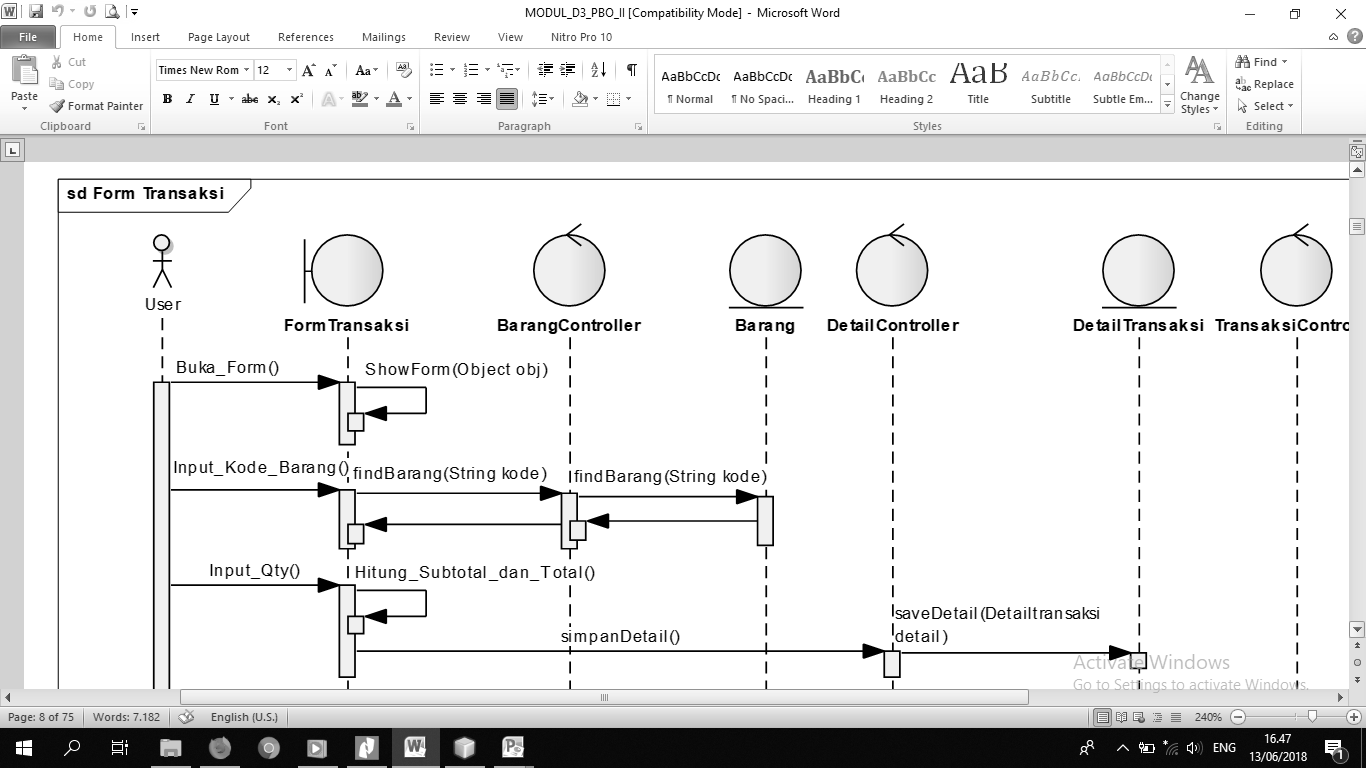


Adalah orang atau sistem yang memperoleh manfaat dari dan berada di luar sistem. Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram.

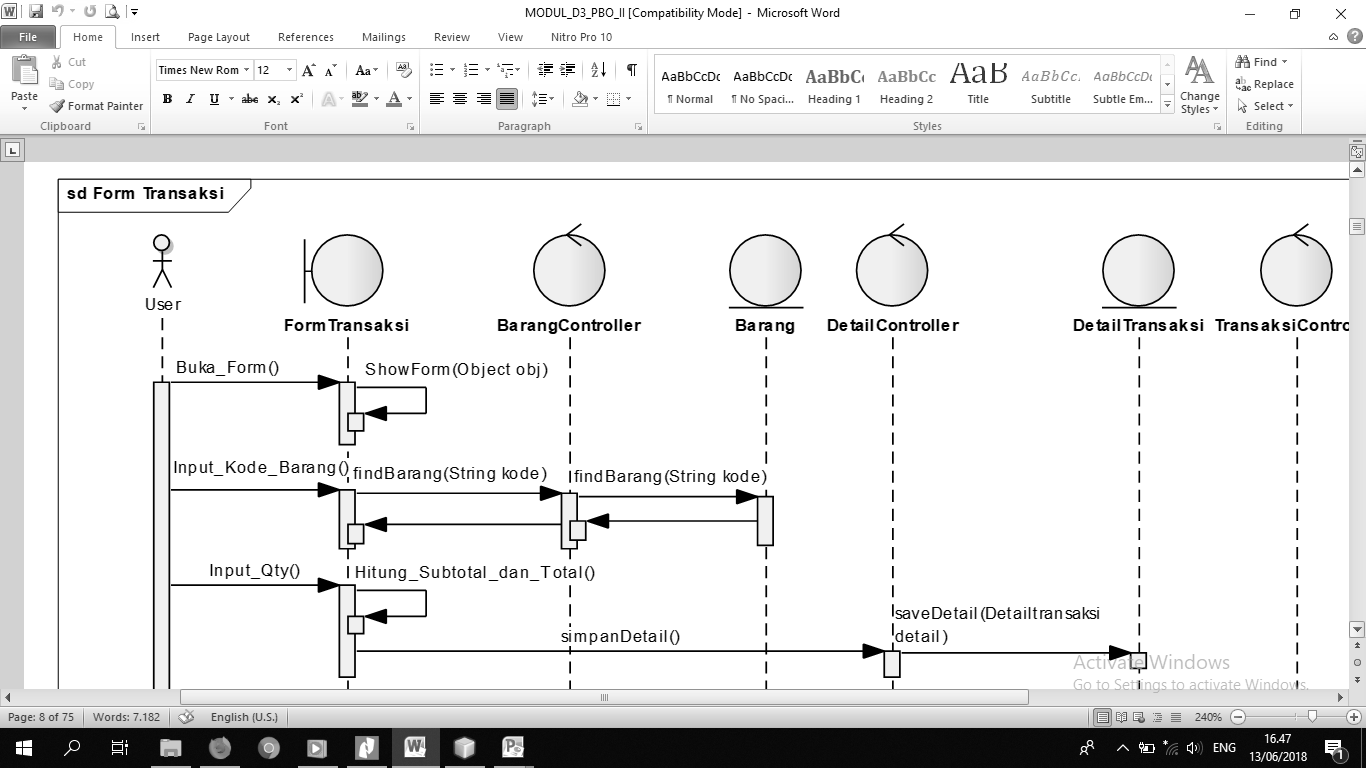
1. *Object*



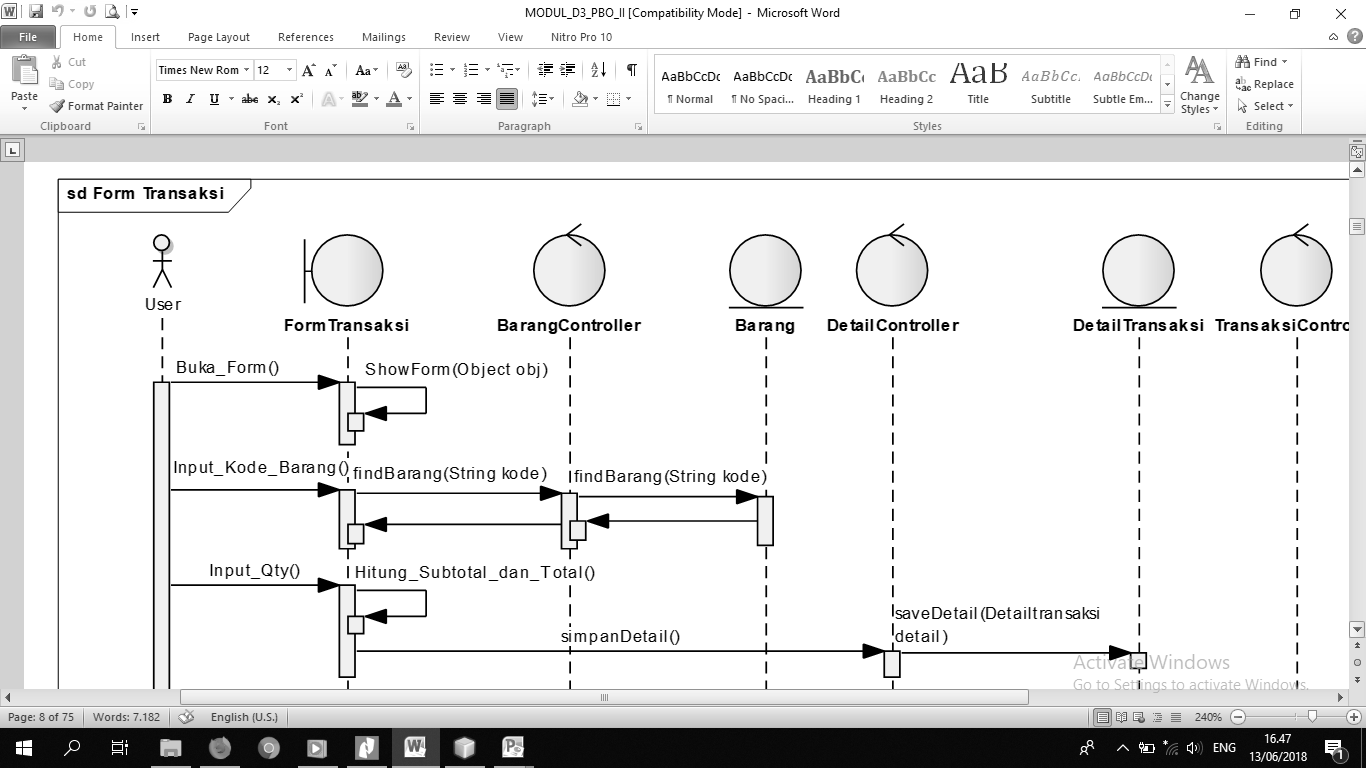
Berpartisipasi dalam suatu urutan dengan mengirim dan / atau menerima pesan. Ditempatkan di bagian atas diagram. Adapun penggambarn lain dari *object* bisa dilihat seperti di bawah ini:



**Boundary Class**: Menggambarkan tampilan program.



**Control Class**: Menggambarkan controller.



**Entity Class:** Menggambarkan class.

1. *Lifeline*



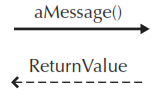
Menunjukkan kehidupan suatu objek selama suatu urutan. Berisi X pada titik di mana kelas tidak lagi berinteraksi.

1. *Execution Occurrence* (Kejadian Eksekusi)



Merupakan persegi panjang sempit panjang yang ditempatkan di atas lifeline. Menunjukkan kapan suatu objek mengirim atau menerima pesan.

1. *Message*



Menyampaikan informasi dari satu objek ke objek lainnya. Pemanggilan operasi diberi label dengan pesan yang dikirim dan panah padat, sedangkan pengembalian diberi label dengan nilai yang dikembalikan dan ditampilkan sebagai tanda panah putus-putus.

1. *Guard Condition*



Merupakan tes yang harus dipenuhi untuk pesan yang akan dikirim.

1. *Object Destruction*



X ditempatkan di ujung lifeline objek untuk menunjukkan bahwa objek tersebut akan keluar dari eksistensi.

1. *Frame*



Menunjukkan konteks sequence diagram.

*Sequence* diagram tidak lepas dari metode MVC. MVC (Model-View-Controller) bisa dijabarkan sebagai berikut:

1. **Model** mewakili struktur data.
2. **View** adalah bagian yang mengatur tampilan ke pengguna.
3. **Controller** merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke tampilan program.

Contoh penerapan MVC sebagai berikut:

1. Sebuah database penjualan memiliki 4 tabel: Barang, User, Transaksi, dan Detail transaksi
2. Masing-masing tabel dibuatkan MVC-nya:
   1. Model berisi variabel sesuai nama field dari tabel beserta fungsi.
   2. View berupa tampilan form program (kecuali detail transaksi).
   3. Controller berisi coding yang berhubungan dengan database.

Contoh dari *sequence diagram* diilustrasikan pada gambar 5.8.



**Gambar 5.8. *Sequence Diagram***

**Penjelasan diagram:**

1. User berinteraksi langsung dengan tampilan layar, pada contoh di atas, User berinteraksi dengan form transaksi. Hal yang pertama dilakukan adalah membuka form transaksi, lalu input kode barang, proses pencarian kode barang melalui controller barang menuju ke entitas Barang, kemudian nama dan harga barang ditampilkan pada form.
2. Selanjutnya User menginput jumlah beli (qty) dan dilakukan perhitungan subtotal dan total. Setelah itu data tersebut masuk disimpan dalam entitas DetailTransaksi melalui controller detail.
3. Setelah semua data diinput, maka semua data transaksi disimpan dalam entitas Transaksi melalui controller transaksi.

## STUDI KASUS

Studi kasus yang akan dibahas pada bab ini adalah mengenai sistem informasi perpustakaan. Sistem pada perpustakaan ini terdiri dari empat prosedur, diantaranya:

1. Pendaftaran Anggota

Seorang calon anggota yang belum menjadi anggota perpustakaan dapat mendaftarkan diri dengan mengisi formulir pendaftaran. Setelah mengisi formulir tersebut, maka data calon anggota dapat diinput ke dalam sistem oleh pustakawan. Pustakawan mencetak kartu anggota, menstempelnya, dan menyerahkan ke anggota perpustakaan.

1. Pengolahan Data Buku

Pustakawan harus memasukkan data koleksi bahan pustaka yang dibelinya ke dalam sistem. Proses pengelolan akan dilaksanakan dan pustakawan dapat mencetak label koleksi untuk masing-masing koleksi bahan pustaka.

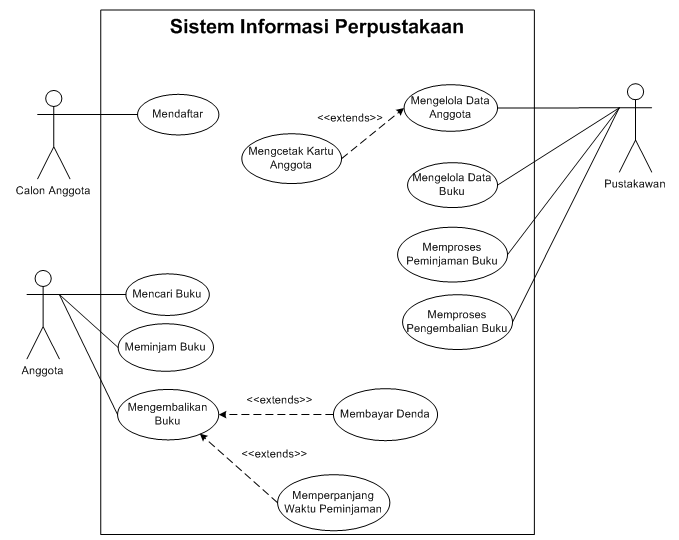
1. Peminjaman Buku

Koleksi bahan pustaka yang telah melalui proses pengolahan, dapat dipinjamkan kepada anggota. Saat seorang anggota meminjam koleksi bahan pustaka, pustakawan harus memasukkan nomor keanggotaan, dan sistem akan menyimpan data peminjaman serta menampilkan form untuk menambahkan data lengkap peminjaman. Pustakawan mengisikan nomor dari koleksi bahan pustaka yang akan dipinjam dan tanggal pengembalian.

1. Pengembalian Buku

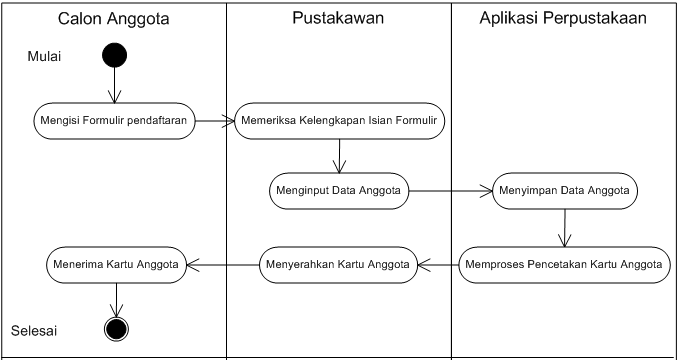
Pustakawan harus mencatat data pengembalian, pada saat anggota mengembalikan buku pinjamannya. Pustakawan mengisikan nomor dari koleksi bahan pustaka yang dikembalikan pada form Pengembalian. Pada form ini terdapat informasi seperti anggota peminjam, koleksi bahan pustaka yang telah dipinjam tapi belum dikembalikan, tanggal pengembalian, denda yang harus dibayar, dan perpanjangan waktu peminjaman.

*Use case diagram* dari studi kasus di atas bisa dilihat pada gambar 5.9.



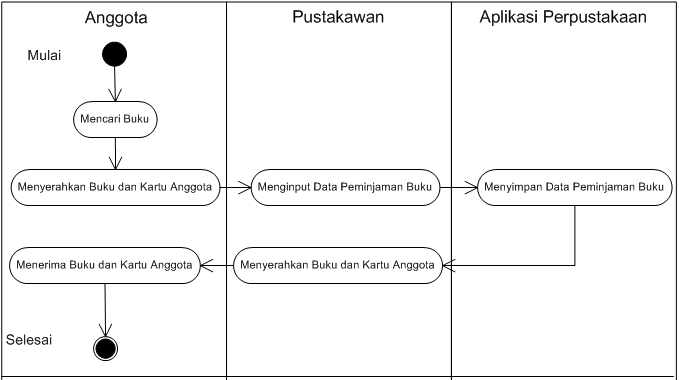
**Gambar 5.9. *Use Case Diagram* Sistem Informasi Perpustakaan**

*Activity diagram* dari *use case* mendaftar bisa dilihat pada gambar 5.10.



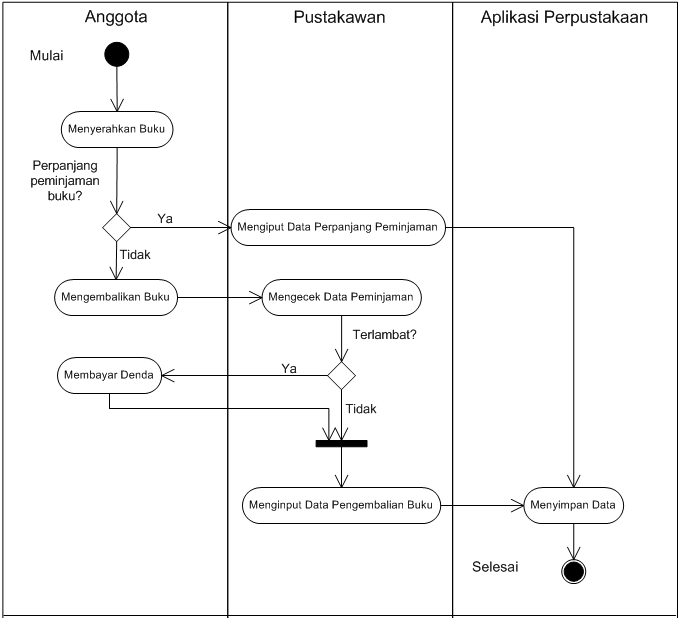
**Gambar 5.10. *Activity Diagram* dari *Use Case* Mendaftar**

*Activity diagram* dari *use case* meminjam buku bisa dilihat pada gambar 5.11.



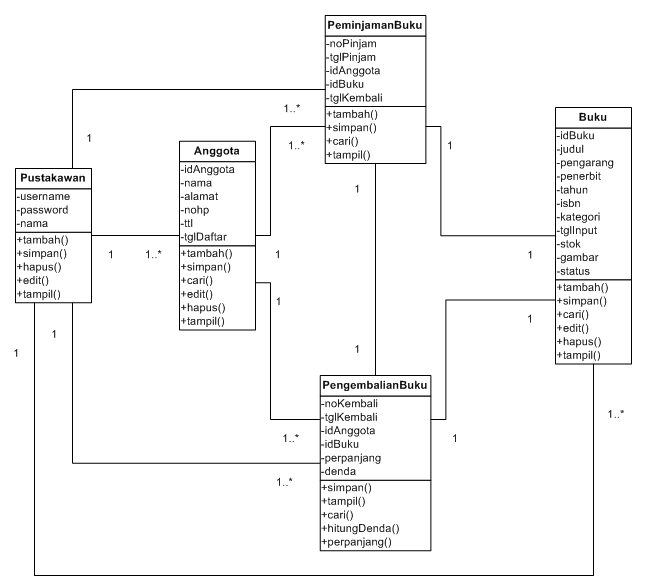
**Gambar 5.11. *Activity Diagram* dari *Use Case* Meminjam Buku**

*Activity diagram* dari *use case* mengembalikan buku bisa dilihat pada gambar 5.12.



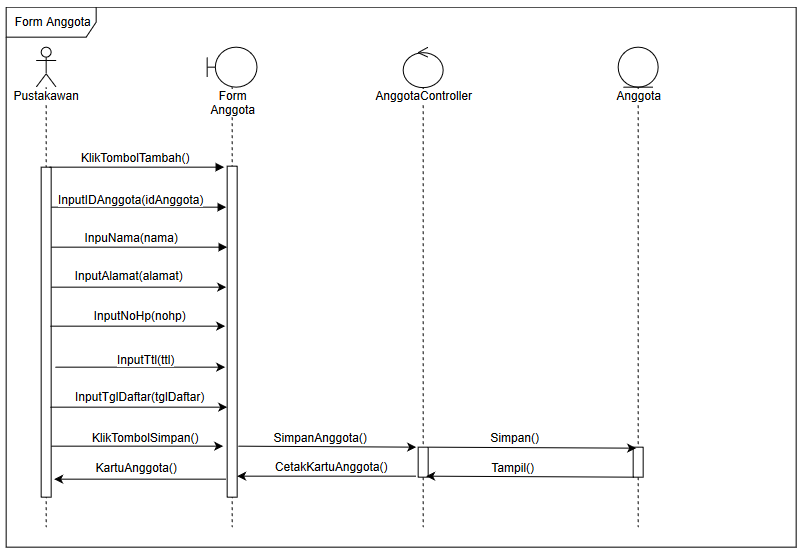
**Gambar 5.12. *Activity Diagram* dari *Use Case* Mengembalikan Buku**

*Class diagram* dari studi kasus di atas bisa dilihat pada gambar 5.13.



**Gambar 5.13. *Class Diagram* Sistem Informasi Perpustakaan**

*Sequence diagram* dari Form Anggota bisa dilihat pada gambar 5.14.



**Gambar 5.14. *Sequence Diagram* Form Anggota**

# BAB VI DESAIN INTERFACE

**Deskripsi :**

Membahas desain user interface, mengetahui kebutuhan output dan input dari yang akan diusulkan dan mahasiswa dapat mendesain dokumen output dan input

**Tujuan Pembelajaran:**

Mahasiswa dapat membuat rancangan input dan output, yang terdiri dari :

1. Perancangan Output

Pedoman perancangan output :

1. Alat perancangan output terinci
2. Mengatur tata letak isi output.
3. Bentuk laporan berbentuk label dan grafik
4. Perancangan input
5. Fungsi dokumen dasar pedoman perancangan input

## 6.1 ANALISA PERANCANGAN INPUT & OUTPUT

Untuk dapat masuk kedalam sistem komputer ada satu bagian yang merupakan penghubung antara user dengan sistem komputer bagian itu adalah yang dikenal dengan interface. Tampilan interface diharapkan dapat menjadi alat berkomunikasi antara user dengan komputer dengan demikian diharapkan tampilan interface dapat mudah dipahami atau yang kita kenal dengan istilah user friendly.

Untuk menghasilkan sebuah tampilan yang mudah dipahami user hendaknya diperhatikan kebutuhan dari dari user dan kebutuhan dari sistem itu sendiri, dengan penyampaian yang baik tentunya masukan dan keluaran yang dihasilkan akan sesuai dengan yang diharapkan. Interaksi yang baik antara user dan komputer diantaranya ditandai dengan adanya respon atau feedback yang menyatakan bahwa masukan yang dibuat oleh user dapat diterima dan diproses oleh sistem dan kondisi lain apabila ternyata terjadi kesalahan maka sistem akan merespon dengan pemberian pesan kesalahan. Untuk itulah perlu diketahui tentang bagaimana cara merancang input dan output yang disesuaikan dengan kebutuhan sistem.

### A. Perancangan Output

sebelum pembahasan tentang bagaimana merancang output ada beberapa hal yang harus anda ketahui tentang bentuk keluaran, diantaranya :

1. Keluaran yang tercetak, memindahkan apa yang ditampilkan di layar ke media kertas atau lainnya melalui perangkat keluaran

2. Tampilan Layar, melalui monitor dapat dilihat hasil proses yang sudah disusun dalam bentuk tabel, gambar, grafik atau desain tampilan keluaran lain

Output adalah bagian hasil dari sebuah sistem komputer, setelah melalui kegiatan input yang dilanjutkan dengan pemrosesan. merancang output dapat berupa teks, tabel atau grafik sebagai laporan yang dapat didokumentasikan dan selanjutnya dapat digunakan dalam pengambilan keputusan

untuk menghasilkan keluaran yang baik ada beberapa hal yang harus diperhatikan diantaranya :

1. mengidentifikasi sistem output dan melakukan review terhadap logical requirements

2. menentukan atau menspesifikasikanresource fisik (device) output

3. lakukan desain untuk preprinted (preview)output yang akan dihasilkan

4. Desain, validasi dan testing output

Output adalah informasi yang dikirim kepada para pengguna melalui sistem informasi dapat berupa hardcopy atau softcopy, melalui internet, ekstranet, atau world wide web.

Tujuan yang harus dicapai penganalis sistem saat merancang output :

1. Merancang output untuk tujuan tertentu
2. Membuat output bermanfaat bagi para pengguna
3. Mengirimi jumlah output yang tepat
4. Menyediakan distribusi output yang tepat
5. Menyediakan output yang tepat waktu
6. Memilih metode output yang paling efektif

Klasifikasi output berdasarkan pendisitribusian dan penerima output menurut (Mulyani, 2016), yaitu :

* 1. Internal Output yaitu output yang ditujukan untuk pemilik sistem atau user yang ada pada internal organisasi
  2. Detail Report, yaitu output yang dihasilkan oleh internal organisasi dengan menyajikan informasi yang selengkap-lengkapnya
  3. Summary Report, yaitu output yang ditujukan untuk level manager sebagai pengambil keputusan berisi rangkuman informasi yang sudah dikelompokkan
  4. Exceptions Report, yaitu output yang berisi hasil filterisasi dengan pengecualian kondisi tertentu
  5. External Output, yaitu output yang menampilkan informasi di luar organisasi
  6. Turnaround Output, yaitu external output yang menjadi input bagi sistem .

Dalam penyusunan prototipe output yang berbasis layar dimana output yang dihasilkan ditampilkan di layar monitor perlu diperhatikan beberapa hal berikut :

1. Menentukan kebutuhan akan laporan tersebut.
2. Menentukan pengguna.
3. Menentukan item-item data yang dimasukan
4. Mengestimasi ukuran laporan secara keseluruhan
5. Judul laporan
6. Nomor halaman
7. Memasukan tanggal persiapan laporan
8. Memberi label setiap kolom data secara tepat.
9. Menentukan data-data varabel
10. Menunjukan tempat baris-baris kosong
11. Mengulas laporan-laporan prototype

Laporan sebagai output yang menampilkan hasil keluaran dari proses dalam sistem memiliki beberapa bentuk, sebagai berikut :

1. Laporan berbentuk tabel

1. Notice Report

Notice report merupakan bentuk laporan yang memerlukan perhatian khusus. Laporan ini harus dibuat sesederhana mungkin, tetapi jelas, karena dimaksudkan supaya permasalahan permasalahan yang terjadi tampak dengan jelas sehingga dapat langsung ditangani.

1. Equiposed Report

Laporan ini biasanya digunakan untuk maksud perencanaan. Dengan disajikannya informasi yang berisi hal-hal bertentangan, maka dapat dijadikan sebagai dasar di dalam pengambilan keputusan

1. Variance Report

Laporan ini menunjukkan selisih (variance) antara standar yang sudah ditetapkan dengan hasil kenyataannya atau sesungguhnya

1. Comparative Report

Isi dari laporan ini adalah membandingkan antara satu hal dengan hal yang lainnya. Misalnya pada laporan rugi/laba atau neraca dapat dibandingkan antara nilai-nilai elemen tahun berjalan dengan tahun-tahun sebclumnya.

1. Laporan Berbentuk Grafis
2. Garis

Pada bagan garis (line chart), variasi dari data ditunjukkan dengan suatu garis atau kurva

1. Batang

Nilai-nilai data dalam bagan batang (bar chart digambarkan dalam bentuk batang-batang vertikal ataupun batang-batang horisontal.

1. Pastel

Bagan pastel (pie chart) merupakan bagan yang berbentuk lingkaran menyerupai kue pastel (pie). Tiap tiap potong dari pie dapat menunjukkan bagian dari data.

### B. Perancangan Input

Dalam pemrosesan data input merupakan bagian yang tidak dapat ditiadakan karena melalui inputan proses bisa berjalan dan menghasilkan keluaran yang diharapkan.

menurut whitten dan Bentley dalam (Mulyani, 2016) dalam melakukan perancangan sistem input ada beberapa hal yang harus diperhatika dan menjadi acuan serta landasan, yaitu :

1. Data Capture

Data Capture merupakan identifikasi atau pengisian data baru. Data capture biasanya menggunakan paper-based. contoh proses data capture adalah pengisian formulir

1. Source Document

Source Document adalah formulir yang digunakan untuk merecord data yang mendeskripsikan suatu transaksi.

1. Data Entry

Data Entry adalah proses konversi atau transalasi data yang berasal dari source document atau sumber data lainnya dalam format yang dimengerti oleh komputer

1. Data Processing

Data Processing merupakan tahapan untuk melakukan pemrosesan data yang sudah diinput.

Dalam mendesain sistem input ada beberapa hal yang harus diperhatikan agar dapat mudah dimengerti user serta meminimalisir kesalahan dalam penginputan data, diantaranya :

1. Penginputan data yang sudah pernah disimpan sebelumnya sebaiknya menggunakan sistem pencarian dan tidak semua harus diinput contohnya untuk menginput data buku setelah menuliskan kodenya apabila kode tersebut sudah tersimpan sebelumnya maka akan langsung muncul datanya sehingga akan mengurangi kemungkinan terjadi kerangkapan data
2. Pemberian pesan kesalahan sehingga data yang tidak sesuai tidak akan tersimpan
3. Hasil perhitungan dimunculkan secara otomatis

Input merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi-transaksi yang dilakukan oleh organisasi. Formulir adalah perangkat penting untuk mengendalikan aliran kerja dan digunakan untuk menangkap (capture) data yang terjadi sering juga disebut Dokumen Dasar.

Tujuan perancangan input :

1. Membuat penyelesaian input yang mudah dan efisien
2. Menjamin input akan memenuhi tujuan yang diharapkan
3. Menjamin penyelesaian yang tepat.
4. Membuat tampilan layar dan formulir yang menarik
5. Membuat input yang tidak rumit
6. Membuat tampilan layar dan formulir yang konsisten

Terdapat beberapa hal yang perlu diperhatikan dari dokumen dasar.

1. Fungsi dokumen dasar
   1. Dapat menunjukan macam dari data
   2. Dapat dicatat dengan jelas, konsisten dan akurat
   3. Bertindak sebagai pendistribusian data
   4. Membantu pembuktian terjadinya transaksi
   5. Dapat digunakan sebagai back up
2. Petunjuk merancang dokumen dasar
   1. Kertas yang digunakan
   2. Ukuran dari dokumen dasar
   3. Warna yang digunakan
   4. Judul dokumen dasar
   5. Nomor dokumen dasar
   6. Nomor urut dokumen dasar
   7. Nomor dan jumlah halaman
   8. Spasi
   9. Pembagian area
   10. Membuat Caption, yaitu box caption, yes no chech off caption, horizontal check off caption, blocked spaces caption, scannable from caption
   11. Instruksi dalam dokumen dasar

Empat garis pedoman untuk merancang formulir

1. Membuat formulir mudah diisi, yaitu dengan memperhatikan aliran formulir, pengelompokan tujuh bagian sebuah formulir, pembuatan judul.
2. Memastikan bahwa formulir akan memenuhi tujuan yang telah dibuat
3. Membuat formulir yang memastikan penyelesaian tepat.
4. Buatlah formulir yang menarik

3. Cara mengurangi masukan

ada beberapa cara yang dapat digunakan untuk mengurangi jumlah masukan, cara yang dapat dilakukan adalah :

1. Menggunakan kode
2. Data yang relatif konstan yang disimpan di file induk
3. Jam dan tanggal dapat diambil dari sistem
4. Rutin perhitungan dilakukan oleh sistem.

## C. Merancang Arsitektur Antarmuka

Ragam aplikasi pada umumnya disajikan dalam beberapa platform yang dibedakan menjadi tiga, yaitu :

* 1. Aplikasi perangkat lunak berbasis web (online)
  2. Aplikasi berbasis desktop (offline)
  3. Aplikasi berbasis mobile

Untuk menghasilkan tampilan antarmuka yang sesuai dibutuhkan sketsa dari rancangan antarmuka tersebut atau yang sering disebut mockup. Mockup digunakan sebagai protitipe arsitektur sistem informasi yang menjelaskan antar muka pemrograman untuk setiap modulnya (Sulianta, 2017)

Ada beberapa aplikasi pembuat mockup yang dapat digunakan, diantaranya :

1. Mockup Builder ([www.mockupbuilder.com](http://www.mockupbuilder.com))
2. Balsamiq Builder ([www.balsamiq.com](http://www.balsamiq.com))
3. Mocking Bird (www. Gomockingbird.com)
4. Mockflow([www.mockflow.com](http://www.mockflow.com))
5. Invision([www.invisionapp.com](http://www.invisionapp.com))

## STUDI KASUS RANCANGAN DOKUMEN INPUT & OUTPUT

Dengan menggunakan studi kasus perpustakaan anda diharapkan dapat merancang tampilan input dan output yang diperlukan dalam sistem informasi perpustakaan, sebelum merancang sebuah aplikasi perlu diketahui kebutuhan sistem yang akan dibuat.

Berikut Rancangan dokumen input & output untuk analisa kebutuhan sistem informasi perpustakaan:

Dokumen Masukan:

1. Formulir Pendaftaran Anggota

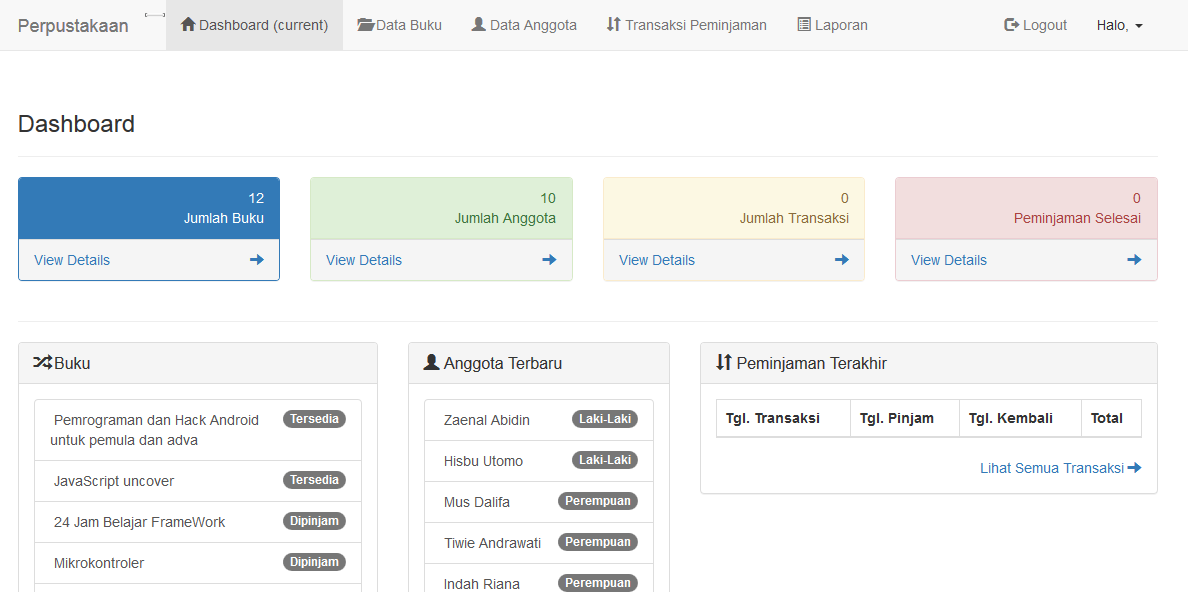
2. Data Buku

Dokumen Keluaran:

1. Kartu Anggota

2. Laporan Peminjaman & Pengembalian Buku

Setelah diketahui kebutuhan sistem maka dilanjutkan dengan merancang tampilan antarmuka, berikut adalah contoh tampilan antar muka untuk aplikasi perpustakaan dimana di dalamnya menampilkan data buku, data anggota serta transaksi peminjaman dan pengembalian serta menu yang menampilkan pilihan untuk pencetakan laporan.

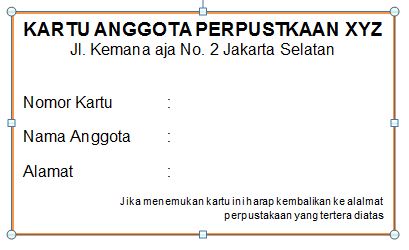


**Gambar 6.1 Antarmuka Aplikasi Perpustakaan**

tampilan berikut adalah antarmuka yang digunakan untuk pengolahan data anggota perpustakaan

**Gambar 6.2 Antarmuka Input Data Anggota**

Sebagai hasil pengolahan data anggota sebagai keluarannya dapat dicetak kartu anggota sebagai bukti bahwa calon anggota tersebut sudah terdaftar menjadi anggota perpustakaan



**Gambar 6.3 Output Kartu Anggota**

setiap transaksi peminjaman buku dan pengembalian buku dicatat dan dilakukan perhitungan apabila ada denda atas keterlambatan pengembalian dan hasil output berupa laporannya dapat ditampilkan seperti di bawah ini :

**Gambar 6.4 Tampilan Report Peminjaman dan Pengembalian Buku**

# 

# BAB VII PENGUJIAN PERANGKAT LUNAK DAN PEMELIHARAAN SISTEM

**Deskripsi :**

Membahas pengujian terhadap program dan pemeliharaan

**Tujuan Pembelajaran:**

Mahasiswa mampu membuat pengujian terhadap program dan melakukan pemeliharaan

## 7.1. Pengujian Perangkat Lunak

Proses menelusuri dan mempelajari sebuah program dalam rangka menemukan kesalahan pada perangkat lunak sebelum diserahkan kepada pengguna (Roger S. Pressman, ). Sasaran pengujian pada perangkat lunak adalah (Sukamto, 2014):

1. Pengujian adalah proses eksekusi suatu program dengan maksud menemukan kesalahan.
2. Test case yang baik adalah test case yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum pernah ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

Karakteristik umum dari pengujian perangkat lunak adalah sebagai berikut :

1. Pengujian dimulai pada level modul dan bekerja kearah integrasi pada sistem berbasiskan komputer.
2. Teknik pengujian yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
3. Pengujian diadakan oleh software developer dan untuk proyek yang besar oleh group testing yang independent.
4. Testing dan debugging adalah aktivitas yang berbeda tetapi debugging harus diakomodasikan pada setiap strategi testing.
5. **Strategi Pengujian Perangkat Lunak**

**Gambar 7.1. Strategi Pengujian Perangkat Lunak**

Strategi pengujian perangkat lunak , yaitu :

1. **Unit testing:** pengujian komponen individual (modul di pemrograman prosedural atau class di OOP).
2. **Integration testing:** pengujian terhadap koleksi dari komponen-komponen yang bekerja bersamaan.
3. **Validation testing:** pengujian aplikasi terhadap kebutuhan pengguna.
4. **System testing:** pengujian aplikasi secara keseluruhan.
5. **Metode Pengujian System Testing**

black-box methods

white-box

methods

Strategies

Methods

**Gambar 7.2. Metode Pengujian System Testing**

1. **White Box Testing**

Pengujian white box adalah pengujian yang didasarkan pada pengecekan terhadap detil perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

**Alasan pengujian White Box:**

1. Adanya kesalahan logik dan asumsi yang tidak tepat pada setiap kemungkinan eksekusi
2. Ada kemungkinan alur program yang tidak tereksekusi
3. Ada kemungkinan kesalahan typographi yang sulit ditemukan kalau tidak dijalankan.

**Kelebihan White Box Testing :**

1. **Kesalahan logika.** Digunakan pada sintaks ‘if’ dan pengulangan. Dimana White Box **Testing** akan mendeteksi kondisi-kondisi yang tidak sesuai dan mendeteksi kapan proses pengulangan akan berhenti.
2. **Ketidaksesuaian asumsi.** Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
3. **Kesalahan ketik.**Mendeteksi bahasa pemrograman yang bersifat case sensitive.

**Kelemahan White Box Testing :**

Untuk perangkat lunak yang tergolong besar, White Box Testing dianggap sebagai strategi yang tergolong boros, karena akan melibatkan sumber daya yang besar untuk melakukannya.

1. **Black Box Testing**

Menurut (Sukamto, 2014), “*Black-box Testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

**KELEBIHAN BLACK BOX TESTING**

1. Dapat memilih subset test secara efektif dan efisien
2. Dapat menemukan cacat
3. Memaksimalkan testing investmen

**KELEMAHAN BLACK BOX TESTING**

Tester tidak pernah yakin apakah Perangkat Lunak tersebut benar – benar lulus uji.

Perbedaan White Box Testing dan Black Box Testing

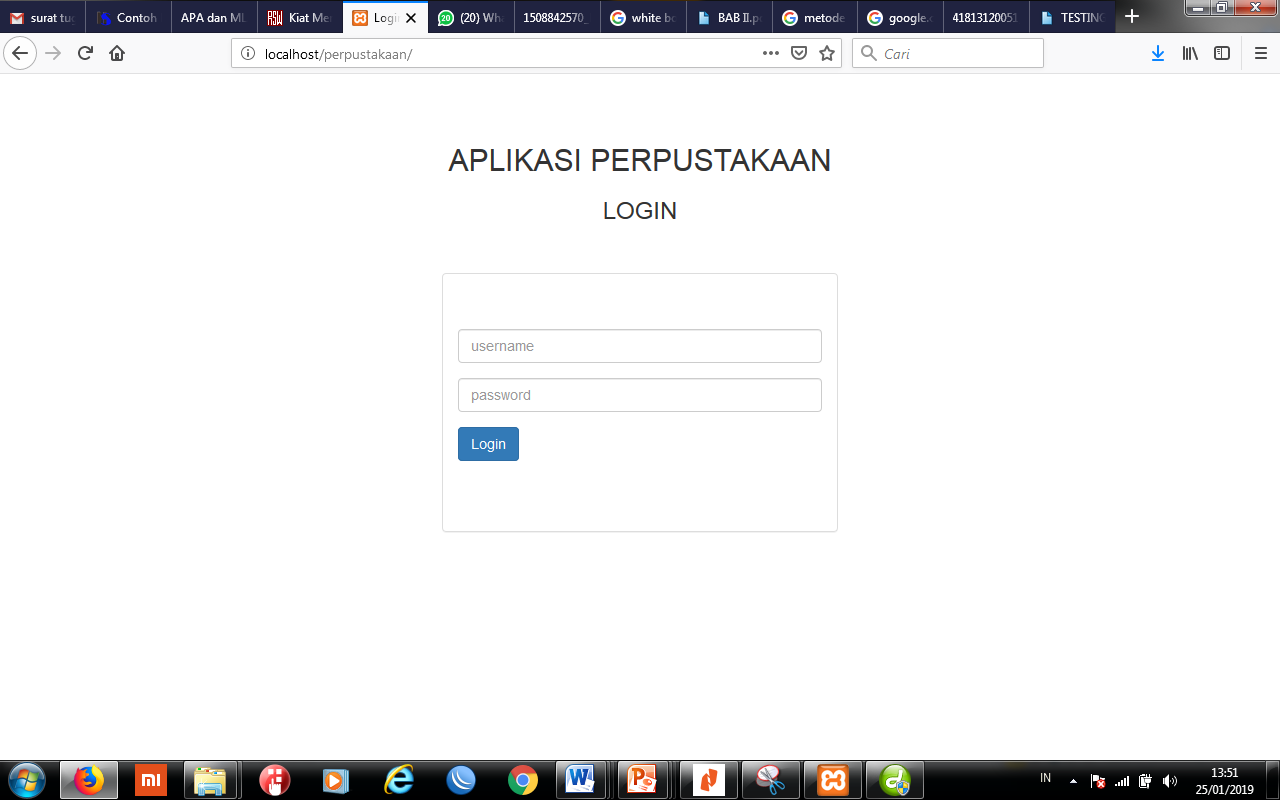
***White box (Struktural)***

1. Dilakukan oleh penguji yang mengetahui tentang QA.
2. Melakukan testing pada software/program aplikasi menyangkut security dan performance program tersebut (meliputi tes code, desain implementasi, security, data flow, software failure).
3. Dilakukan seiring dengan tahapan pengembangan software atau pada tahap testing.

***Metode BlackBox  (Fungsional)***

1. Dilakukan oleh penguji Independent.
2. Melakukan pengujian berdasarkan apa yang dilihat, hanya fokus terhadap fungsionalitas dan output. Pengujian lebih ditujukan pada desain software sesuai standar dan reaksi apabila terdapat celah-celah bug/vulnerabilitas pada program aplikasi tersebut setelah dilakukan white box testing.
3. Dilakukan setelah white box testing.

Contoh Pengujian Black Box Testing



**Gambar 7.3. Halaman Login Aplikasi Perpustakaan**

**Tabel 7.1. Contoh Pengujian Halaman Login Admin**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Skenario Pengujian | *Test Case* | Hasil yang di harapkan | Hasil Pengujian | Kesimpulan |
| 1. | Mengetikan *username* dan *password* tidak di isi kemudian klik tombol masuk | *Username*:  (*admin*)  *Password*:  (kosong) | Sistem akan menolak lalu muncul pesan “Periksa *username* dan *password* anda” | Sesuai dengan harapan | *Valid* |
| 2. | Mengetikan *username* tidak diisi dan *password* di isi kemudian klik tombol masuk | *Username:*  (kosong)  *Password:*  (*admin*) | Sistem akan menolak lalu muncul pesan “Periksa *username* dan *password* anda” | Sesuai dengan harapan | *Valid* |
| 3. | Mengetikan *username* dengan benar dan mengisi *password* salah kemudian klik tombol masuk | *Username*:  (*admin*)  (benar)  *Password* :  (maruko)  (salah) | Sistem akan menolak lalu muncul pesan “Periksa *username* dan *password* anda” | Sesuai dengan harapan | *Valid* |
| 4. | Mengetikan *username* dan *password* dengan data yang benar kemudian klik tombol masuk | *Username*:  *Admin* (benar)  *Password*:  *Admin* (benar) | *Login* berhasil dan Akan muncul halaman *home admin* | Sesuai dengan harapan | *Valid* |

## 7.2. Pemeliharaan Sistem

Sistem perlu dipelihara karena beberapa hal, yaitu:

1. Sistem memiliki kesalahan yang dulunya belum terdeteksi, sehingga kesalahan-kesalahan sistem perlu diperbaiki.
2. Sistem mengalami perubahan-perubahan karena permintaan baru dari pemakai sistem.
3. Sistem mengalami perubahan karena perubahan lingkungan luar (perubahan bisnis).
4. Sistem perlu ditingkatkan.

Jenis Pemeliharaan Sistem :

Pemeliharaan sistem dapat digolongkan menjadi empat jenis:

**Pemeliharaan Korektif**

Pemeliharaan korektif adalah bagian pemeliharaan sistem yang tidak begitu tinggi nilainya dan lebih membebani, karena pemeliharaan ini mengkoreksi kesalahan-kesalahan yang ditemukan pada saat sistem berjalan.

**Pemeliharaan Adaptif**

Pemeliharaan adaptif dilakukan untuk menyesuaikan perubahan dalam lingkungan data atau pemrosesan dan memenuhi persyaratan pemakai baru.

**Pemeliharaan Penyempurnaan**

Pemeliharaan penyempurnaan mempertifi cara kerja atau maintainabilitas (kemampuan untuk dipelihara).

**Pemeliharaan Preventif**

Pemeliharaan preventif terdiri atas inspeksi periodik dan pemeriksaan sistem untuk mengungkap dan mengantisipasi permasalahan.

Contoh Pemeliharaan Sistem

**1. Spesifikasi *Hardware***

a. *Server*

1) *CPU*

(a) *Processor Pentium® Core 2 Duo*

(b) *RAM DDR2 4 GB*

(c) *Hard Disk 500 GB*

2) *Mouse*

3) *Keyboard*

4) *Monitor dengan resolusi layar minimum 1024x768*

5) *Koneksi internet dengan kecepatan 2 Mbps.*

b. *Client*

1) *CPU*

(a) *Processor Pentium® 4*

(b) *RAM DDR2 1GB*

(c) *Hard Disk 20 GB*

2) *Mouse*

3) *Keyboard*

4) *Monitor dengan resolusi layar minimum 1024x768*

5) Koneksi *internet dengan kecepatan 56 kbps.*

**2. Spesifikasi *Software***

1. *Server*

a. Sistem operasi yang umum digunakan seperti: *Microsoft Windows atau Linux*

(*Ubuntu, Fedora, dll).*

b. *Aplikasi bundle web server seperti: Xampp, WampServer, php2triad yang terdiri*

dari beberapa komponen, diantaranya:

(1) Aplikasi *Apache Server v2*

(2) Aplikasi *PHP Server v5*

(3) Aplikasi *MySQL Server v5*

(4) Aplikasi *phpMyAdmin v3*

c. Aplikasi *Web seperti Mozilla Firefox, Opera, Safari, Internet Explorer,*

*Google Chrome.*

*2. Client*

a. Sistem operasi yang umum digunakan seperti: *Microsoft Windows atau Linux*

(*Ubuntu, Fedora, dan lain-lain).*

b. Aplikasi w*eb browser seperti Mozilla Firefox, Opera, Safari, Internet Explorer,*

*Google Chrome.*

# DAFTAR PUSTAKA

Antonius. (2010). *Petunjuk Menyusun Karya Tulis Ilmiah*. Bandung: Sarana Tutorial Nurani Sejahtera.

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *Systems Analysis & Design An Object-Oriented Approach with UML* (5th Editio). Unitetd States of America: Wiley.

Fathansyah. (2018). *Basis Data* (Revisi ke). Bandung: Informatika.

Hasibuan, Z. A. (2017). *Metodologi Penelitian Pada bidang Ilmu Komputer Dan Teknologi Informasi*. Fakultas ilmu Komputer Universitas Indonesia.

Limbong, T. (2012). *Diktat Metode Penelitian IT*. Medan: STMIK Budidarma.

Mc Leod, R., & Schell, G. . (2007). *Management Information System*. Philadelphia: The Dryden Press.

Mulyani, S. (2016). *Metode Analisis dan Perancangan Sistem Edisi Ke 2 Cetakan Pertama*. Bandung: Abdi Sistematika.

Sugiarto. (2017). *Metodologi Penelitian Bisnis*. Yogyakarta: ANDI.

Sukamto, R. A. dan M. S. (2014). *Rekayasa Perangkat Lunak Terstruktur dan berorientasi objek*. Informatika Bandung.

Sulianta, F. (2017). *Teknik Perancangan Arsitektur Sistem Informasi*. Yogyakarta: ANDI.

# BIODATA

***Fintri Indriyani, M.Kom*,** Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Pendidikan terakhir adalah Magister Manajemen Sistem Informasi di STMIK Nusa Mandiri Jakarta. Menyelesaikan pendidikan Strata 1 (S1) di Sekolah Tinggi Manajemen Informatika dan Komputer Kuwera, Program Studi Manajemen Informatika tahun 2005. Penulis telah tersertifikasi dosen tahun 2015 dan pernah menjadi redaksi jurnal Paradigma.

***Ynuita M.Kom*,** Penulis adalah staf pengajar di Universitas Bina Sarana Informatika. Lahir di Jakarta tanggal 24 Juni 1983. Pendidikan terakhir adalah Magister Manajemen Sistem Informasi di STMIK Nusa Mandiri Jakarta lulus tahun 2013. Menyelesaikan pendidikan Strata 1 (S1) di Sekolah Tinggi Manajemen Informatika dan Komputer Swadarma, Program Studi Manajemen Informatika lulus tahun 2008.

***Dinda Ayu Muthia, M.Kom,***lahir di Jakarta 8 Juli 1988, adalah dosen tetap Universitas hBina Sarana Informatika, Fakultas Teknologi Informasi, Jurusan Sistem Informasi untuk matakuliah Pemrograman Berorientasi Objek dan Pemodelan Sistem Berorientasi Objek. Menyelesaikan pendidikan Strata 1 (S1) STMIK Nusa Mandiri, Jurusan Sistem Informasi Tahun 2011; Pendidikan Strata 2 (S2) Pascasarjana STMIK Nusa Mandiri, Program Studi Ilmu Komputer Konsentrasi Sistem Informasi Manajemen Tahun 2013. Selain sebagai staf pengajar, penulis juga pernah menjadi kontributor dalam penulisan artikel dengan segmen teknologi di majalah Pesona BSI selama tahun 2015-2016.

***Artika Surniandari, M.Kom,*** lahir di Depok, 06 Oktober 1981, adalah dosen tetap Universitas Bina Sarana Informatika, Fakultas Teknologi Informasi, Program Studi Sistem Informasi, Menyelesaikan pendidikan Strata 1 (S1) di Sekolah Tinggi Manajemen Informatika dan Komputer Kuwera, Program Studi Sistem Informasi tahun 2006; Strata 1 (S1) Sekolah Tinggi Manajemen Informatika dan Komputer Kuwera, Program Studi Sistem Informasi Tahun 2011; Pendidikan Strata 2 (S2) Pascasarjana STMIK Nusa Mandiri, Program Studi Ilmu Komputer Konsentrasi Sistem Informasi Manajemen Tahun 2010. Selain sebagai staf pengajar, penulis juga pernah menjadi redaksi di majalah Pesona BSI selama tahun 2013-2015*.*

***Sriyadi, M.Kom***. Tahun 2005 lulus pendidikan Program Strata Satu (S1) Program Studi Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Nusa Mandiri Jakarta. Tahun 2011 lulus Program Strata Dua (S2) Pascasarjana STMIK Nusa Mandiri dengan konsentrasi bidang Manajemen Information System (MIS). Aktifitas sebagai dosen Universitas Bina Sarana Informatika. Karya peneltian: Sistem Informasi E-Kepegawaian Pada PT. Mannapratama Santosa Jakarta terbit pada jurnal Information System For Educators And Professionals, Solusi Cerdas Menyusun Laporan Pertanggungjawaban Koperasi Dengan Penerapan E-Koperasi diterbitkan pada jurnal Ilmu Ekonomi dan Pembangunan, Sistem Informasi Penerimaan Peserta Didik Baru (PPDB) Online Pada Madrasah Aliyah Negeri 17 Jakarta pada Seminar Nasional Inovasi dan Tren (SNIT) 2018, Pemakalah pada Forum International Conference On Islamic s and Philantropy (ICIEBP). Sebagai pembina Gerakan Mahasiswa Berprestasi Program Studi Sistem Informasi Universitas Bina Sarana Informatika