

BAB II

KAJIAN LITERATUR

2.1 Konsep Sistem Informasi

2.1.1 Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Kata sistem berasal dari Bahasa Latin (*Systema*) dan bahasa Yunani (*Sustema*) yang berarti sebuah kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan [4].

Beberapa karakteristik yang harus dimiliki sebuah sistem, yaitu [4]:

1. Memiliki Komponen

Komponen ini merupakan bagian dari sebuah sistem interaksi, dimana keseluruhan komponen tersebut saling berinteraksi satu sama lain.

2. Memiliki Batasan atau *Boundary*

Batasan ini merupakan pembatas dari sebuah sistem informasi tersebut menjadi satu buah kesatuan sistem informasi yang utuh, dan menunjukkan ruang lingkup yang dimiliki oleh sistem informasi tersebut.

3. Memiliki *Environment*

Environment merupakan keseluruhan sistem dan juga lingkungan yang berada di luar batasan atau *boundary* dari sistem informasi.

4. Memiliki *Interface*

Interface atau antarmuka ini merupakan media yang digunakan untuk dapat menghubungkan sebuah komponen atau sub-sistem yang terdapat pada sebuah sistem informasi.

5. Memiliki *Input* atau Masukan Sistem

Input sistem atau sistem masukan ini merupakan jenis energi yang digunakan untuk dimasukkan ke dalam suatu sistem. Masukan atau *input* ini terdiri dari dua jenis, yaitu:

- a. *Maintenance Input* merupakan *input* yang berhubungan dengan perawatan suatu sistem, dimana merupakan sebuah energi yang dimasukkan ke dalam sistem informasi, agar sistem informasi tersebut bisa berjalan dengan baik dan optimal.
- b. *Signal Input* merupakan energi yang merupakan sinyal, yang artinya, energi ini sangat berpengaruh terhadap proses transfer dan juga transmisi data atau

informasi yang dimiliki sebuah *host* untuk diteruskan melalui sistem informasi menuju keluaran atau *output*.

6. Memiliki *Output* atau Keluaran Sistem

Output atau keluaran merupakan karakteristik dari sistem informasi yang berikutnya. *Output* merupakan keluaran energi atau hasil yang diteruskan oleh *input*. Hasil atau *output* ini bisa berupa tampilnya data dan juga informasi yang muncul pada *display user*

7. Memiliki Pengolah dan Pemrosesan Sistem

Pengolah data atau pemrosesan sistem ini merupakan komponen atau bagian di dalam sebuah sistem informasi yang memiliki tugas utama untuk memproses *input* dari sebuah sistem informasi menjadi keluaran atau *output* dari sebuah sistem informasi. Singkatnya, *processing* sistem ini membantu proses pengolahan data secara keseluruhan yang ada di dalam sebuah sistem informasi, lalu mentransmisikan hasil dari pengolahan data tersebut menuju *output* yang dikeluarkan oleh sistem dan dapat diakses oleh *user*.

8. Memiliki Sasaran Sistem

Sasaran sistem merupakan analisis berupa siapa saja yang akan menggunakan sistem informasi ini. Tanpa adanya sasaran dari pembuatan sistem, maka sudah pasti sebuah sistem informasi tidak akan bisa bermanfaat dan juga berguna.

2.1.2 Data

Data merupakan kumpulan fakta atau sesuatu digunakan sebagai *input* yang dioleh dalam proses dan akan menghasilkan sistem informasi (*output*). Sebuah data harus sesuai kebenarannya (*reliable*), akurat, tepat waktu dan mencakup ruang lingkup yang luas [5].

Beberapa fungsi atau manfaat data, yaitu [5]:

1. Bahan dasar pembuat keputusan

Dalam pembuat keputusan terbaik diperlukan data pada suatu permasalahan yang ada. Dengan adanya data membuat keputusan akan lebih mudah dan dapat dipertanggung jawabkan.

2. Bahan dasar perencanaan

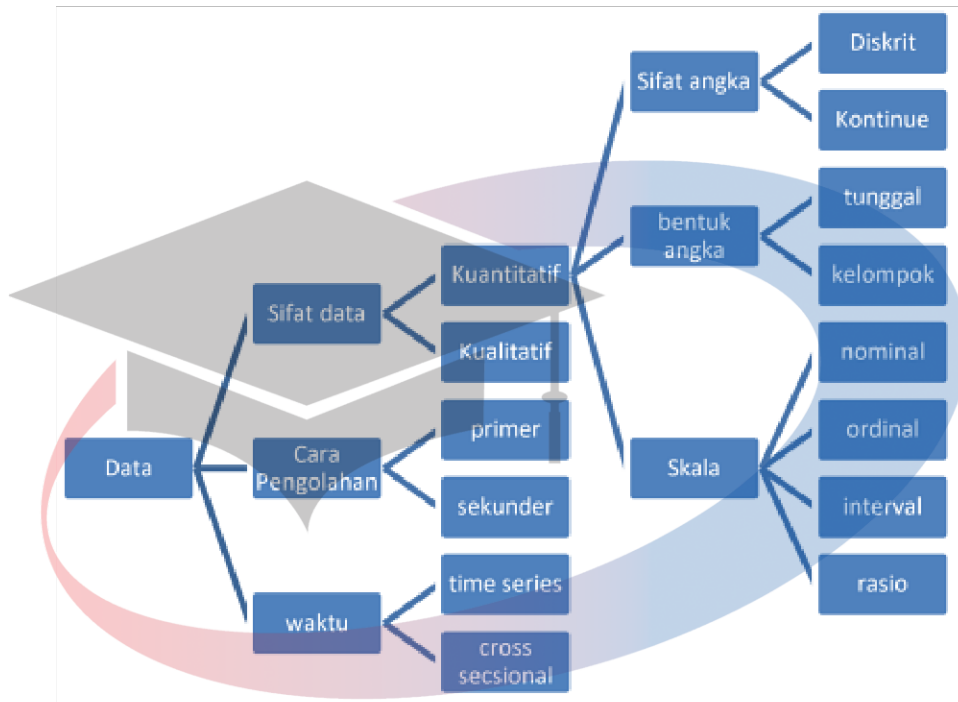
Data dapat digunakan sebagai parameter yang kuat dalam membuat suatu perencanaan. Dengan adanya data, perencanaan dapat lebih terarah dan tingkat keberhasilan dalam mencapai tujuan akan lebih tinggi.

3. Bahan acuan implementasi suatu kegiatan

Dalam suatu kegiatan, data digunakan untuk tolak ukur dalam melakukan kegiatan tersebut. Data yang terintegrasi dapat memudahkan pelaksanaan kegiatan, sehingga kegiatan dapat lebih terstruktur.

4. Bahan evaluasi

Di suatu organisasi data dijadikan bahan evaluasi dalam peningkatan mutu organisasi tersebut.

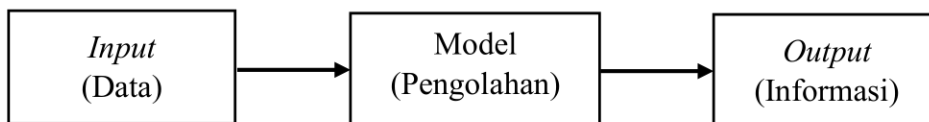


Gambar 2.1 Contoh Jenis-jenis Data

2.1.3 Informasi

Informasi merupakan hasil pengolahan data dengan cara tertentu sehingga lebih berarti dan berguna bagi penerimanya [5].

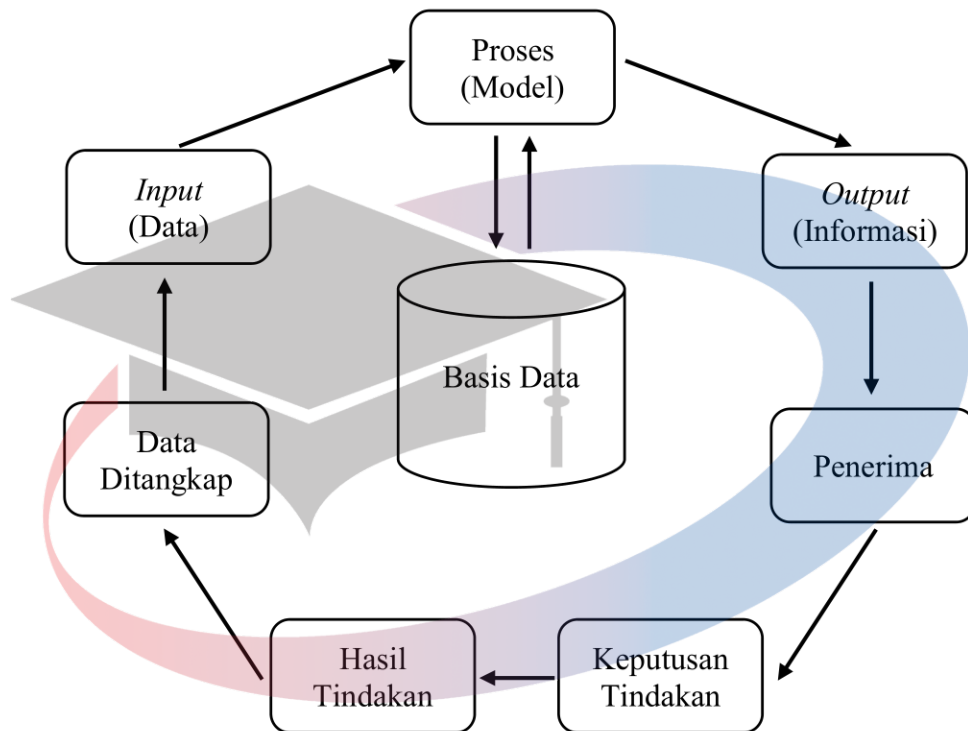
Sumber dari informasi berupa data yang menggambarkan kejadian secara nyata yang telah terjadi pada saat tertentu. Sumber ini perlu diolah melalui sebuah siklus yang dinamakan sebagai siklus pengolahan data (*Data Processing Life Cycle*) [5].



Gambar 2.2 Siklus Pengolahan Data

Pada data (*input*) untuk menghasilkan informasi data diolah sehingga mendapatkan *output*. Dalam pengolahan suatu data diperlukan model tertentu sehingga menjadi informasi yang dapat bermanfaat bagi penerima dalam mengambil keputusan maupun

melakukan kegiatan dan evaluasi. Data yang belum diolah akan disimpan yang bentuknya berupa basis data. Data penyimpanan ini dapat diambil lagi ketika akan diolah menjadi informasi. Data tersebut sebagai *input*, diproses menggunakan model, sehingga menghasilkan *output* dan ditangkap oleh penerima dalam membuat keputusan dan melakukan tindakan dan seterusnya membentuk sebuah siklus yang disebut siklus informasi (*Information Cycle*) [5].



Gambar 2.3 Siklus Informasi

2.1.4 Sistem Informasi

Sistem informasi merupakan sebuah alat atau sarana yang bertujuan untuk mengelola data menjadi informasi, yang dapat dimanfaatkan oleh pengambil keputusan [4].

Komponen-komponen sistem informasi sebagai berikut [4]:

1. Blok Masukan (*Input Block*)

Blok masukan dalam sebuah sistem informasi meliputi metode-metode dan media untuk menangkap data yang akan di masukan dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok model ini terdiri dari kombinasi prosedur logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan *toolbox* dari pekerjaan sistem informasi, teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*). Teknologi blok adalah komponen yang membantu mempercepat proses yang terjadi dalam sistem.

5. Blok Basis Data (*Database Block*)

Database Adalah kumpulan data yang berhubungan satu sama lain, disimpan dalam perangkat keras komputer dan perangkat lunak komputer untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi Kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat yang disebut dengan *DBMS (Data Base Management sistem)*.

6. Blok Kendali (*Controls Block*)

Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya. Meliputi masalah pengendalian terhadap operasional sistem yang berfungsi mencegah dan menangani kesalahan dan kegagalan sistem. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila telanjur terjadi kesalahan dapat langsung cepat diatasi.

2.2 *User Interface (UI)*

User Interface (UI) adalah saat sistem dan pengguna dapat saling berinteraksi satu dengan lainnya melalui perintah seperti halnya menggunakan konten dan memasukkan data [6].

User Interface (UI) adalah tampilan *visual* dari desain sistem sebagai bagian dari UX. Tampilan ini memungkinkan pengguna untuk terhubung dan berinteraksi dengan

produk. Selain digunakan sebagai penyambung, UI juga digunakan untuk mempercantik tampilan sehingga dapat meningkatkan kepuasan pengguna. Namun, tidak hanya harus cantik, tapi UI juga harus mudah digunakan. Beberapa komponen UI termasuk komponen tombol, ikon cetak, tema, tata letak, animasi yang muncul, dan efek visual interaktif lainnya. Desain semua komponen UI ini berfokus pada estetika dan kemudahan penggunaan [7].

Seorang desainer UI akan membuat desain yang memudahkan pengguna programnya. Adapun desain itu disesuaikan dengan tingkat kebutuhan dasar pengguna terhadap program aplikasi web maupun *mobile* tersebut. *Output* dari hasil desainer UI ialah program dengan segala fitur yang sesuai dengan kebutuhan pengguna dalam menggunakan program tersebut [8].

2.3 User Experience (UX)

User Experience (UX) adalah pengalaman yang terkait dengan reaksi, persepsi, perilaku, emosi dan pikiran pengguna saat menggunakan sistem [6].

User Experience (UX) adalah proses mendesain produk melalui metode pengguna. Dengan metode ini, dapat membuat produk yang sesuai dengan kebutuhan pengguna. Produk dengan desain UX yang baik akan memberikan pengalaman yang menyenangkan bagi pengguna menjadi rileks dan nyaman. Pada saat yang sama, komponen UX ini mencakup fungsi-fungsi yang disediakan pada produk, struktur desain, navigasi produk, aspek desain visual, dan semua aspek interaksi dengan pengguna. UX juga memperkenalkan bagaimana mendefinisikan merek, konten dan *copywriting* sesuai dengan target audiensnya [7].

Seorang desainer UX akan merancang program aplikasi web atau *mobile* berdasarkan pengalaman dari pengguna. Dengan begitu, program yang dirancangnya menjadi lebih mudah digunakan oleh penggunanya [8].

Tabel 2.1 Perbedaan UI dan UX

Perbedaan	<i>User Interface (UI)</i>	<i>User Experience (UX)</i>
Tujuan	Membuat tampilan produk lebih menarik.	Memberikan kenyamanan saat memakai produk.
Fokus	Berfokus pada tampilan yang bagus.	Berfokus pada kenyamanan pengguna.
Elemen	Animasi, <i>typography</i> , warna, video, <i>buttons</i> , dan	<i>Usability</i> , navigasi, struktur desain, fitur-fitur,

	lain sebagainya.	<i>interaction design</i> , dan lain sebagainya.
<i>Based On</i>	Desain berdasarkan riset desain dan konsep produk.	Desain berdasarkan riset pengguna.
<i>Tools</i>	Spesifikasi utama desain grafis, seperti Flinto, <i>Principle</i> , <i>Frame X</i> , <i>Adobe Illustrator</i>	Spesifikasi utama <i>prototyping</i> seperti <i>Sketch</i> , <i>InVision</i> , <i>Figma</i> , <i>Adobe XD</i> , <i>Axure</i> .
<i>Skill yang dibutuhkan</i>	Desain grafis, <i>creative thinking</i> , <i>convergent</i> , dan desain <i>branding</i>	Riset, <i>critical thinking</i> , <i>creative thinking</i> , <i>analysis</i> , <i>problem solving</i> , dan <i>wirefarming</i>

2.4 Penjualan

Penjualan adalah suatu proses dimana penjual memenuhi segala kebutuhan dan keinginan pembeli agar dicapai manfaatnya bagi penjual maupun pembeli yang berkelanjutan dan menguntungkan bagi kedua belah pihak [9].

Sales Person atau *salesman* adalah individu yang menawarkan suatu produk dalam proses penjualan [10].

Langkah-langkah dalam melakukan proses penjualan, yaitu [10]:

1. *Approach* (Pendekatan Kepada Prospek)

Pendekatan kepada calon pembeli atau konsumen memerlukan persiapan dan perencanaan yang baik, antara lain mengetahui tentang siapakah calon pembeli atau konsumen, apakah kebutuhan dan keinginannya, adakah kemungkinan perubahan situasi atas kebutuhan maupun produk yang ditawarkan, dan siapkan kiat dengan penolakan atau keberatan.

2. *Presentation* (Penyajian)

Dalam tahapan presentasi, seorang *sales* harus sanggup menjual produk dan membantu memuaskan kebutuhan para konsumen, memperhatikan hal-hal penting, seperti tata bahasa yang baik, kesopanan, jelas atau tepat, tidak berbelit-belit, dan memberikan kesan pertama yang positif.

3. *Selling* (Menjual)

Proses menjual dari seorang *sales* harus dapat memberikan keyakinan kepada pembeli atas manfaat dan kelebihan produk yang ditawarkan. Dengan bekal *product knowledge*

dan pengembangan komunikasi yang efektif, diharapkan calon pembeli menjadi pelaku pembeli.

4. *Closing The Sale* (Menutup Penjualan)

Suatu transaksi penjualan terjadi karena penjualan memperoleh persetujuan dari calon pembeli untuk membeli atau menggunakan produk atau jasa yang ditawarkan. Tahapan ini merupakan keberhasilan seorang penjual memengaruhi dan meyakinkan calon pembeli.

5. *After Sales Service* (Pelayanan Setelah Penjualan)

Kesempurnaan dari suatu penjualan yang berhasil adalah ditutup dengan pelayanan purna jual. Kegiatan tersebut antara lain pemberian ucapan terima kasih melalui surat atas pembelian suatu produk atau jasa, memberikan hadiah, dan mengadakan *direct contact* secara reguler.

2.5 Pemasaran

Pemasaran adalah fungsi organisasi dan serangkaian proses untuk menciptakan, mengkomunikasikan, dan memberikan nilai kepada pelanggan untuk mengelola hubungan pelanggan dengan cara yang menguntungkan organisasi dan pihak-pihak yang berkepentingan terhadap organisasi [11].

Pemasar (*marketer*) adalah seseorang yang mencari respons, perhatian, pembelian, dukungan, dan sumbangan dari pihak lain yang disebut prospek (*prospect*) [11].

Ada 8 keadaan permintaan yang mungkin terjadi, yaitu [11]:

1. Permintaan Negatif

Semua atau sebagian terbesar dari segmen pasar potensial yang penting tidak menyukai produk atau jasa yang ditawarkan, bahkan mereka bersedia membayar untuk menghindarinya.

2. Tidak Ada Permintaan

Orang yang tidak berminat sama sekali terhadap penawaran suatu produk atau jasa.

3. Permintaan *Latent*

Sebagian besar orang-orang mempunyai kebutuhan yang kuat akan sesuatu yang tidak ada dalam bentuk barang atau jasa.

4. Permintaan Menurun

Suatu keadaan dimana permintaan untuk suatu produk atau jasa itu semakin berkurang dari tingkat sebelumnya, dan diperkirakan akan menurun terus jika tidak dilakukan

usaha-usaha untuk memperbaiki pasar yang dituju, penawaran, dan usaha-usaha pemasaran

5. Permintaan Tidak Teratur

Suatu keadaan dimana pola permintaan pada saat-saat tertentu dipengaruhi oleh fluktuasi musim atau hal-hal lain.

6. Permintaan Penuh

Suatu keadaan dimana tingkat dan saat permintaan yang sekarang sama dengan tingkat dan saat permintaan yang diharapkan.

7. Permintaan Berkelebihan

Suatu keadaan permintaan lebih besar dari penawarannya, dengan kata lain bahwa barang yang diminta konsumen lebih banyak dibanding barang yang dipasarkan.

8. Permintaan Tidak Sehat

Suatu jenis produk atau jasa yang permintaannya dinilai kurang baik dari segi kesejahteraan konsumen, kemakmuran masyarakat atau penyedia.

2.6 Laundry

Jasa *laundry* merupakan sebuah bisnis dibidang jasa yang melayani cuci pakaian atau barang lainnya yang umum digunakan konsumen yang biasanya menggunakan mesin cuci, mesin pengering, dan cairan pembersih atau pewangi [12]. Untuk itu usaha ini tidak lapuk oleh zaman, karena tanpa disadari usaha jasa *laundry* sudah menjadi bagian hidup penting manusia dan akan tetap ada [13].

Di dalam suatu usaha terdapat suatu bentuk standar operasi prosedur (*SOP*) sebagai berikut [14].

Untuk pencucian dibedakan menjadi dua, yaitu [14]:

1. Pencucian kiloan adalah pencucian dengan menggunakan dasar perhitungan dari berat timbangan pakaian. Keunggulan dari pencucian kiloan yaitu harganya yang terjangkau dan proses pencucian dapat dilakukan dengan cepat, pencucian kiloan terdiri dari cuci komplit, cuci saja, setrika, keringkan, dan cuci tidak di campur.
2. Pencucian khusus adalah pencucian dengan perhitungan potongan pakaian, dengan perhatian khusus disesuaikan bahan dan jenis pakaian sesuai *washing care* label sebagai petunjuk pencucian yang tertera pada label dalam pakaian. Untuk mengetahui *symbol* atau label yang tertera di dalam pakaian yaitu pakaian akan lebih awet dan terjaga warna aslinya walau telah dicuci berkali-kali, dan selalu dalam keadaan rapi karena setiap satu jenis pakaian diberi hanger dan plastik sendiri. Perlunya sistem

pemisahan pakaian dalam pencucian kiloan bertujuan untuk mempermudah dan lebih mengefektifkan di dalam proses pengerjaan. Kapasitas mesin untuk satu kali mencuci yaitu 5 kg pakaian. Tentunya konsumen memberikan order tidak semua genap 5 kg. Oleh karena itu, agar dalam proses pencucian dapat lebih efisien, pakaian konsumen disatukan untuk mendapatkan jumlah 5 kg. Namun pakaian yang disatukan rawan terjadi risiko tertukar antar pakaian dan terjadi kelunturan. Untuk mengatasi maka dibuat sistem sebagai berikut:

- a. Pemberian nomor dan penembakan top pin
 - b. Pemisahan pakaian luntur
 - c. Penggabungan pakaian
3. Setelah proses pencucian dilanjutkan dengan proses pengeringan. Pakaian keluar dari mesin cuci telah kering 80%, untuk proses selanjutnya pakaian dikeringkan menggunakan *dryer* agar kering 100%.
 4. Setelah proses pengeringan dilakukan, proses setrika pakaian merupakan bagian penting dalam penggarapan proses *laundry*. Agar proses menyetrika lebih efisien pisahkan bahan-bahan sejenis agar suhu setrika tidak sering di ubah dan pemanasan setrika berurutan dari dingin, hangat, dan sampai panas.
 5. Setelah proses setrika selesai dilanjutkan dengan proses penyemprotan pewangi dan pengepakan. Proses pengepakan diawali dahulu dengan pakaian tebal dan bagian bawah lalu semakin ke atas pakaian lebih tipis. Setelah pakaian tertata rapi baru dilanjutkan dengan proses *packing* ke dalam plastik. Setelah proses *packing*, dilanjutkan pemberian nama, nomor nota, dan jumlah pakaian dengan spidol *permanent* pada permukaan atas pada plastik. Letakkan cuci kiloan yang sudah di *packing* dan di tempel nota pada rak, urutkan penempatan dari nomor kecil ke nomor besar.

2.7 Rapid Application Development

Rapid Application Development (RAD) adalah model pengembangan perangkat lunak sekuensial linear yang menekankan siklus pengembangan yang sangat singkat (sekitar 60-90 hari) [15].

RAD juga merupakan suatu pendekatan berorientasi objek terhadap pengembangan perangkat lunak yang bertujuan untuk mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara desain dan implementasi sistem informasi [15].

Model *Rapid Application Development* terdiri atas 3 fase, yaitu [16]:

1. *Requirement Planning*

Dalam tahap ini diketahui apa saja yang menjadi kebutuhan sistem yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alternatif pemecahan masalah, Analisis digunakan untuk mengetahui perilaku sistem dan juga untuk mengetahui aktivitas apa saja yang ada dalam sistem tersebut

2. *Design Workshop*

Mengidentifikasi solusi alternatif dan memilih solusi yang terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapatkan dan dimodelkan dalam arsitektur sistem informasi. *Tools* yang digunakan dalam pemodelan sistem biasanya menggunakan *unified modeling language (UML)*.

3. *Implementation*

Setelah *design workshop* dilakukan, selanjutnya sistem diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program. Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan.



Gambar 2.4 Tahapan-tahapan dalam Model RAD

2.8 Teknik Perancangan Sistem

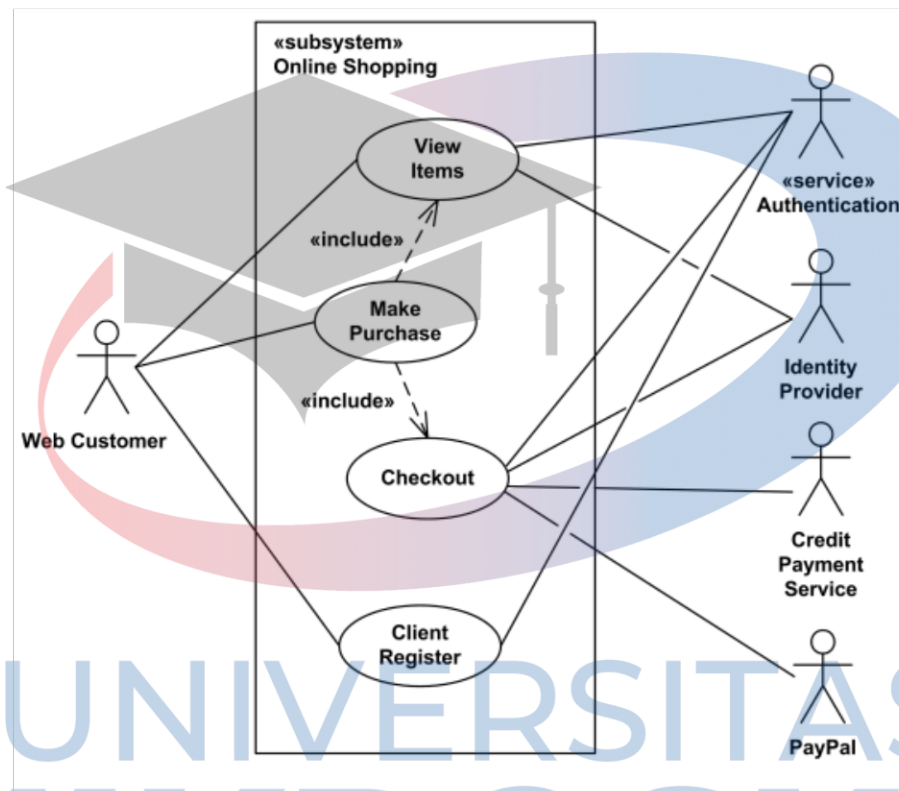
2.8.1 Use Case Diagram

Use case diagram adalah diagram yang digunakan untuk menggambarkan hubungan antara sistem dengan aktor. Diagram ini hanya menggambarkan secara global, karena *use case diagram* hanya menggambarkan sistem secara global, maka elemen-elemen yang digunakan pun sangat sedikit [17].

Use case diagram adalah bentuk utama dari persyaratan sistem atau perangkat lunak untuk program perangkat lunak yang baru berkembang. *Use case* menentukan perilaku yang diharapkan (apa), dan bukan metode yang tepat untuk mewujudkannya (bagaimana) [18].

Tujuan dari *use case diagram*, yaitu [18]:

1. Memetakan kebutuhan sistem.
2. Merepresentasikan interaksi pengguna terhadap sistem
3. Untuk mengetahui kebutuhan di luar sistem.

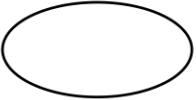






Gambar 2.5 Contoh penggambaran *Use Case Diagram*

Berikut merupakan tabel yang menjelaskan keterangan dari simbol yang terdapat pada *Use Case Diagram* [17].

Tabel 2.2 Simbol-simbol pada *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .

	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar proses antar unit atau aktor.
	<i>Extend</i>	<i>Use case</i> yang berbeda menangani pengecualian dari <i>use case</i> dasar. Panah dari di perpanjang ke <i>use case</i> dasar.
	<i>Include</i>	<i>Use case</i> berisi perilaku yang umum untuk lebih dari satu <i>use case</i> lainnya. Panah menunjukkan ke <i>use case</i> umum.
	<i>Association</i>	Menghubungkan satu objek dengan objek lainnya.
	<i>Generalizes</i>	Sebuah aktor lebih bersifat umum daripada aktor lainnya. Simbol panah menunjukkan ke aktor umum.

Use case description berfungsi untuk melengkapi diagram *use case* agar lebih dipahami konteksnya. Disajikan dalam bentuk tabel, *use case description* adalah gambaran secara general mengenai fungsionalitas proses bisnis berupa skenario yang melibatkan berjalannya suatu sistem. Ada 13 komponen yang terdapat pada *use case description* untuk menjelaskan masing-masing *use case* secara lengkap. Berikut adalah penjelasan dari komponen yang terdapat pada *use case description* tersebut [19].

No.	Komponen	Deskripsi
1.	<i>Use Case Id</i>	Berisikan kode unik untuk membedakan setiap <i>use case</i> .
2.	<i>Use Case Name</i>	Berisikan nama <i>use case</i> .
3.	<i>Description</i>	Berisikan penjelasan singkat mengenai fungsi <i>use case</i> .
4.	<i>Actor</i>	Berisikan informasi mengenai pengguna atau aktor yang terlibat pada <i>use case</i> tersebut.
5.	<i>Pre Condition</i>	Berisikan kondisi yang harus ada atau sudah terjadi sebelum <i>use case</i> dijalankan. Prasyarat apa yang harus dipenuhi sebelum <i>use case</i> dijalankan.
6.	<i>Post Condition</i>	Proses yang dihasilkan dari kegiatan <i>use case</i> .

7.	<i>Basic Flow</i>	Berisi penjelasan langkah-langkah normal yang berakhir sukses. Ditandai dengan awal, <i>body</i> , dan akhir.
8.	<i>Exceptions Flow</i>	Menjelaskan informasi mengenai kendala-kendala yang menyebabkan skenario dasar tidak dapat dipenuhi.
9.	<i>Variations</i>	Berisi mengenai tindakan alternatif apabila ada pengecualian dari skenario dasar. Merupakan representasi notasi <i><extend></i> pada diagram <i>use case</i> .
10.	<i>Extensions</i>	Berisi mengenai skenario tambahan yang merujuk pada notasi <i><include></i> dan biasanya diidentifikasi apabila ada langkah-langkah yang terlibat tetapi tidak terikat dengan arus konteks skenario awal.
11.	<i>Business Rules</i>	Berisi tentang aturan bisnis yang terdapat pada <i>use case</i> .
12.	<i>Non Functional Requirement</i>	Penyampaian kebutuhan diluar fungsi <i>use case</i> dapat disampaikan disini. Umumnya berisi kinerja, kapasitas, dan keterbatasan sistem.

2.8.2 Activity Diagram

Activity diagram adalah bagian penting dari *UML* yang menggambarkan aspek dinamis dari sistem. Logika prosedural, proses bisnis dari aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity diagram* [20].

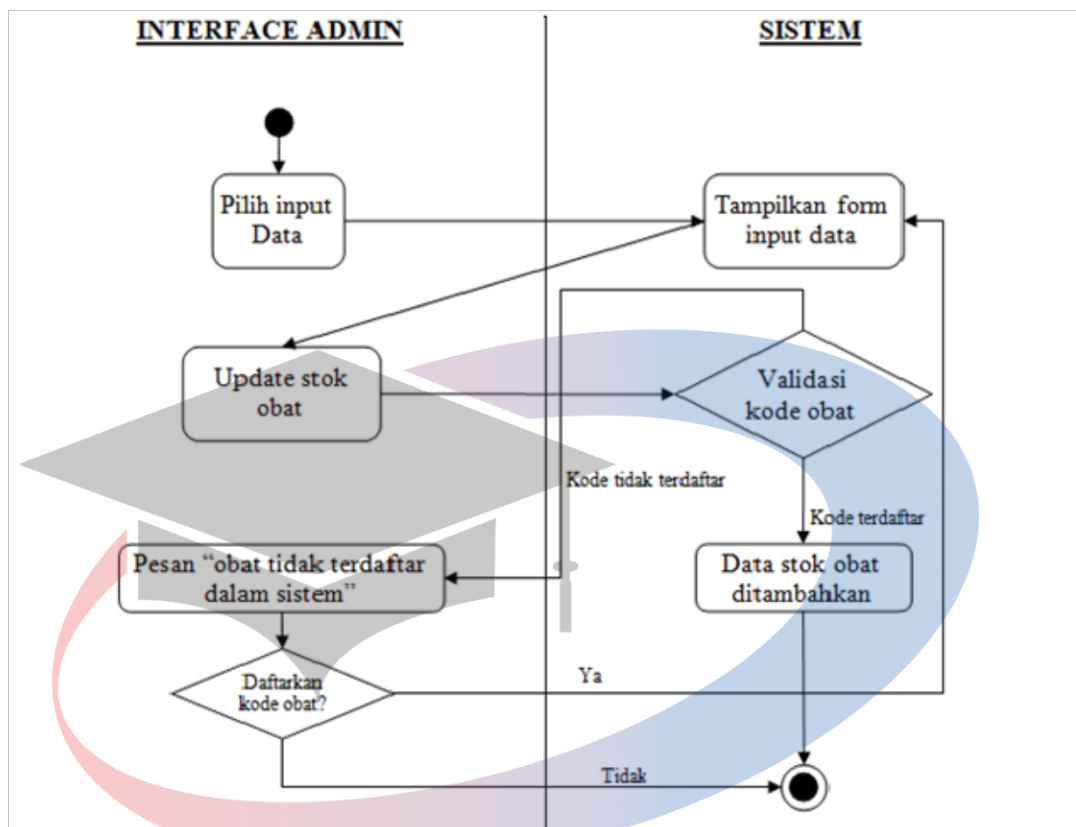
Activity diagram adalah diagram yang digunakan untuk menggambarkan alur kerja (aktivitas) pada *use case* (proses), logika, proses bisnis, dan hubungan antara aktor dengan alur-alur kerja *use case* [17].

Activity diagram menggambarkan aliran kontrol dari sistem target, seperti mengeksplorasi aturan bisnis yang kompleks dan menggambarkan proses bisnis [18].

Tujuan dari *activity diagram*, yaitu [16]:

1. Menggambarkan suatu model data untuk sistem informasi, tidak peduli apakah model data tersebut sederhana ataupun kompleks.
2. Untuk meningkatkan pemahaman mengenai gambaran umum skema dari suatu aplikasi.
3. Mampu menyatakan secara visual akan kebutuhan spesifik suatu informasi serta dapat menyebarkan informasi tersebut ke seluruh bisnis.

4. Dapat membuat bagan secara jelas dan terperinci dengan cara memperhatikan kode spesifik apa yang dibutuhkan suatu program sehingga mampu mengimplementasikannya ke struktur yang digambarkan.



Gambar 2.6 Contoh penggambaran *Activity Diagram*

Berikut merupakan tabel yang menjelaskan keterangan dari simbol yang terdapat pada *Activity Diagram* [17].

Tabel 2.3 Simbol-simbol pada *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Status awal aktivitas sistem.
	<i>Activity</i>	Aktivitas yang dilakukan sistem.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Fork</i>	Untuk memecah <i>behaviour</i> menjadi <i>activity</i> atau <i>action</i> yang paralel

	<i>Join</i>	Untuk menggabungkan kembali <i>activity</i> atau <i>action</i> yang paralel
	<i>Final Node</i>	Status akhir yang dilakukan sistem.

2.8.3 *Sequence Diagram*

Sequence diagram adalah penggambaran kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek [21].

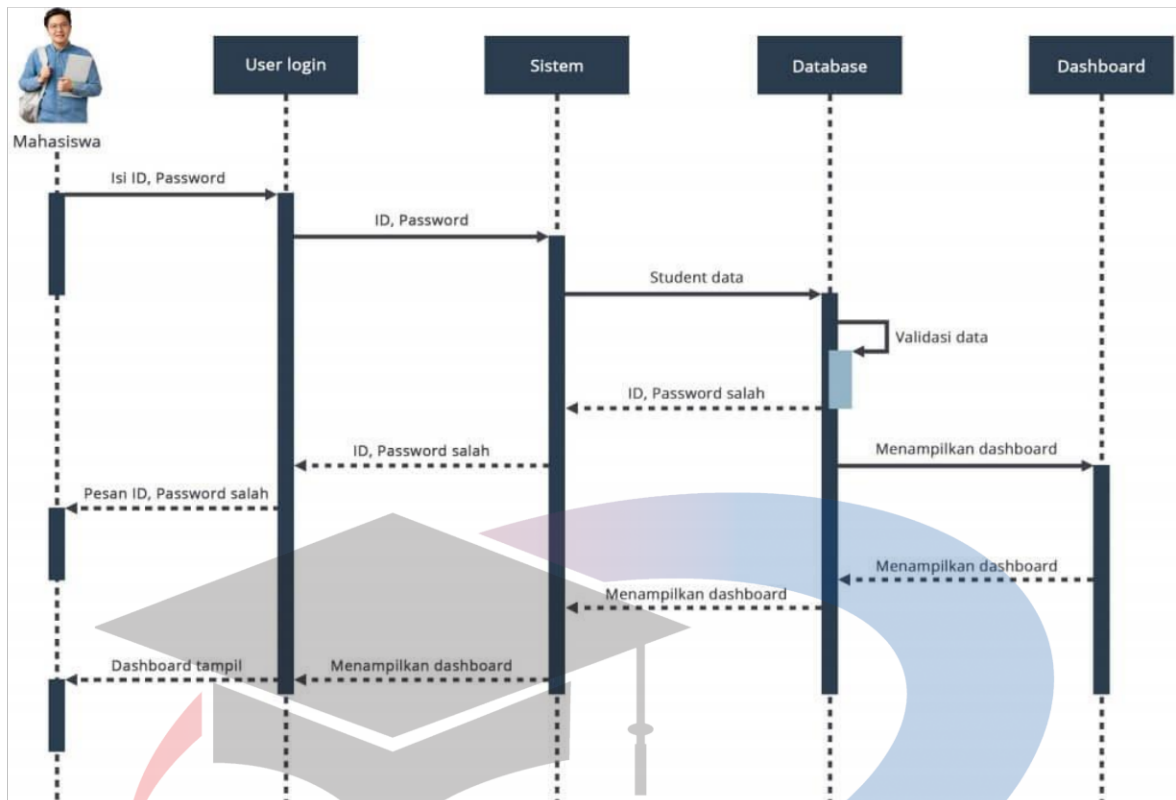
Sequence diagram digunakan untuk menggambarkan perilaku sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini di dalam *use case* [20].

Sequence diagram memodelkan kolaborasi objek berdasarkan urutan waktu, sehingga menunjukkan bagaimana objek berinteraksi dengan orang lain dalam skenario tertentu dari *use case* [18].

Tujuan dari *sequence diagram*, yaitu [18]:

1. Model interaksi tingkat tinggi antara objek aktif dalam suatu sistem.
2. Model interaksi antara *instance* objek dalam kolaborasi yang mewujudkan *use case*.
3. Model interaksi antara objek dalam kolaborasi yang mewujudkan operasi.

UNIVERSITAS
MIKROSKIL



Gambar 2.7 Contoh penggambaran *Sequence Diagram*

Berikut merupakan tabel yang menjelaskan keterangan dari simbol yang terdapat *Sequence Diagram* [21].

Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>Entity</i> antarmuka yang saling berinteraksi.
	<i>Object Message</i>	Komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
	<i>Self Message</i>	Komunikasi kepada diri sendiri yang memuat informasi-informasi tentang aktivitas yang terjadi.

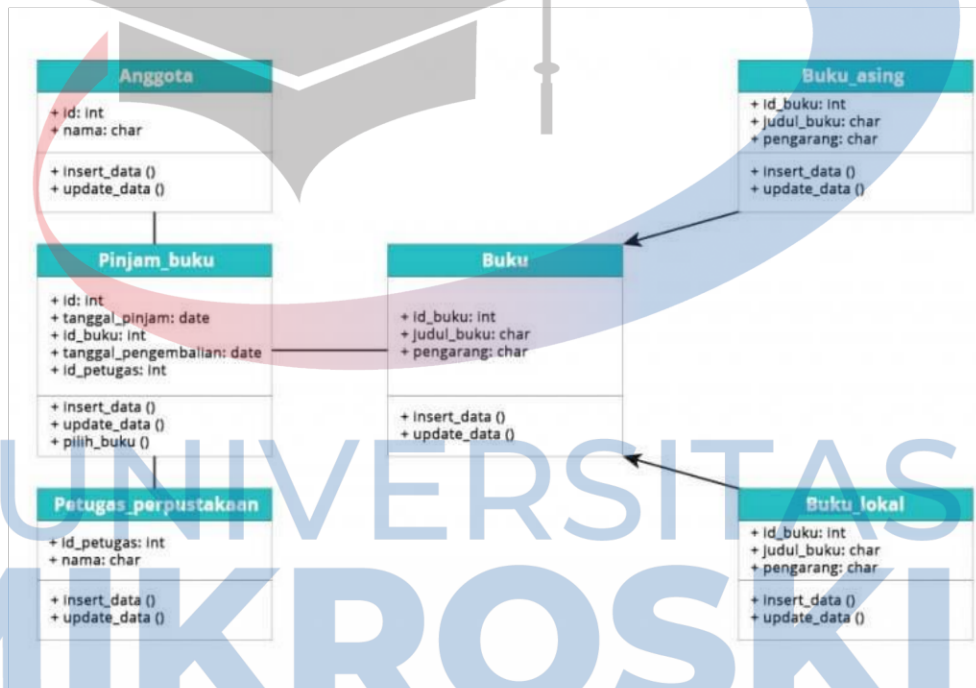
2.8.4 *Class Diagram*

Class diagram adalah suatu model statis yang menunjukkan *class-class* dan hubungannya di antaranya dan senantiasa konstan di dalam sistem sepanjang waktu. *Class Diagram* menggambarkan *class* beserta perilaku dan keadaan yang menghubungkannya antar *class-class* [22].

Class diagram merupakan teknik pemodelan sentral yang berjalan melalui hampir semua metode berorientasi objek. Diagram ini menjelaskan jenis objek dalam sistem dan berbagai macam hubungan statis yang ada di antara objek [18].

Tujuan dari *class diagram*, yaitu [18]:

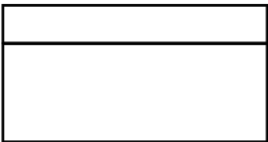
1. Menunjukkan struktur statis pengklasifikasi dalam suatu sistem.
2. Diagram menyediakan notasi dasar untuk diagram struktur lain yang ditentukan oleh *UML*.
3. Dapat digunakan untuk memodelkan sistem dari perspektif bisnis.

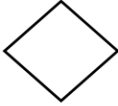

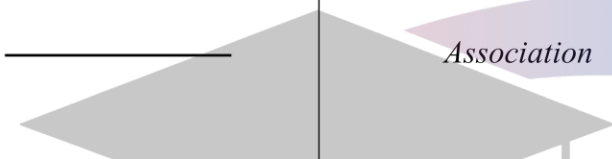

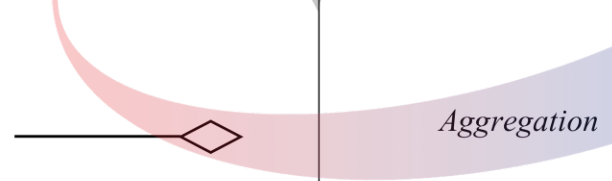



Gambar 2.8 Contoh penggambaran *Class Diagram*

Berikut merupakan tabel yang menjelaskan keterangan dari simbol yang terdapat pada *Class Diagram* [22].

Tabel 2.5 Simbol-simbol pada *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	Himpunan objek-objek yang berbagi atribut serta operasi yang sama.

	<p><i>Nary Association</i></p>	<p>Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.</p>
	<p><i>Collaboration</i></p>	<p>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.</p>
	<p><i>Association</i></p>	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain.</p>
	<p><i>Generalizes</i></p>	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
	<p><i>Aggregation</i></p>	<p>Suatu jenis asosisasi yang salah satu ujungnya dibagikan sebagai jenis agregasi, yang berarti memiliki agregasi bersama.</p>
	<p><i>Composition</i></p>	<p>Sebuah bentuk agregasi yang kuat yang membutuhkan <i>istance part</i> dimasukkan dalam paling banyak satu komposit sekaligus. Jika komposit dihapus, semua bagiannya biasanya dihapus bersamanya.</p>