BAB 2

TINJAUAN PUSTAKA

2.1 Distro

Berikut ini adalah pengertian dari Distro sebagai tempat penelitian untuk membangun Aplikasi Panduan Belanja Distro di Kota Cimahi.

2.1.1 Pengertian Distro

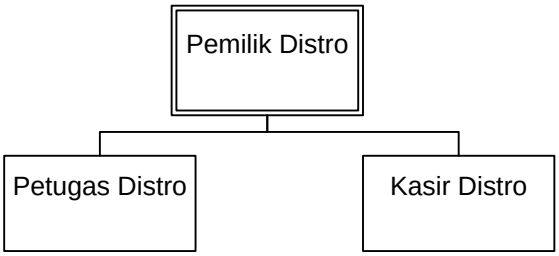
Distro, singkatan dari *distribution store* adalah jenis toko di Indonesia yang menjual pakaian dan aksesori yang dititipkan oleh pembuat pakaian, atau diproduksi sendiri[2]. Distro umumnya merupakan industri kecil dan menengah (IKM) yang sandang dengan merk independen yang dikembangkan kalangan muda. Produk yang dihasilkan oleh distro jumlahnya sangat terbatas, tujuannya agar mempertahankan sifat eksklusif suatu produk dan hasil kerajinan distro tersebut.

2.1.2 Sejarah Distro

Konsep awal sejarah distro dimulai pada pertengahan 1990-an di Bandung. Pada saat itu band-band independen di Bandung berusaha menjual *merchandise* mereka seperti CD/kaset, t-shirt, dan sticker selain di tempat mereka melakukan pertunjukan. Bentuk awal distro adalah usaha rumahan dan dibuat etalase dan rak untuk mempermudah dalam penjualan *t-shirt*. Selain komunitas musik, banyak juga komunitas lain yang mengikuti untuk menjual *merchandise* yaitu seperti komunitas punk dan skateboard yang kemudian juga membuat toko-toko kecil untuk menjual pakaian dan aksesori *merchandise* produk mereka. Kini, industri distro sudah berkembang, bahkan dianggap sudah berhasil menghasilkan produk-produk yang memiliki kualitas ekspor. Seiring dengan perkembangannya baju, kaos dan merchandise tersebut banyak diminati orang, sehingga dibentuklah *distribution store* yang populer dengan sebutan Distro. Pada tahun 2007 diperkirakan ada sekitar 700 unit usaha distro di Indonesia, dan 300 diantaranya ada di Bandung[2].

2.1.3 Struktur Organisasi

Struktur organisasi adalah suatu susunan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi secara umum struktur organisasi distro di kota Cimahi dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Struktur Organisasi Umum Distro Kota Cimahi

Dalam studi kasus yang diteliti terdapat 4 distro untuk di jelaskan susunan struktur organisasinya. Adapun bagian-bagian pekerjaan yang terdapat pada distro sebagai berikut :

1. Distro ORDR
    1) Pemilik Distro : Febrianus Alexander
    2) Petugas Distro : Shendy
    3) Kasir Distro : Shendy
    4) Alamat distro : Jl. Gandawijaya no. 112 Cimahi
2. Distro Shut Up
    1) Pemilik Distro : Junaedi
    2) Petugas Distro : Alex
    3) Kasir Distro : Najib
    4) Alamat Distro : Jl. Gandawijaya no.68 Cimahi
3. Distro Pandemonium
    1) Pemilik Distro : Junaedi

2) Petugas Distro : Yudha
3) Kasir Distro : Rika
4) Alamat Distro : Jl. Gandawijaya no. 141 Cimahi

4. Distro Kandang Babi 31
   1) Pemilik distro : Francis
   2) Petugas Distro : Francis
   3) Kasir Distro : Francis
   4) Alamat Distro : Jl. Warung contong no.31 rt/rw. 01/04 Cimahi

2.1.4 Deskripsi Tugas

Berikut adalah deskripsi tugas dari struktur organisai tiap distro adalah sebagai berikut :

1. Distro ORDR
   1) Pemilik Distro
      a. Bertanggung jawab atas keseluruhan yang ada di distro ORDR
      b. Bertugas untuk salah satu tugas dari kasir dan petugas atau dua-duanya apabila pemilik distro dan atau salah satu pegawai tidak hadir
      c. Bertugas menerima laporan stok barang dari petugas distro
      d. Menerima laporan pemasukkan dan pengeluaran keuangan dari kasir distro.
   2) Petugas Distro
      a. bertugas mencari ketersediaan stok barang untuk customer.
      b. Menampilkan barang pada toko distro ORDR.
      c. Menyimpan stok barang yang tersedia di distro.
      d. Memberikan laporan stok barang distro ke pemilik distro.
   3) Kasir Distro
      a. Bertugas membuat 2 struk pembelian untuk customer dan arsip distro.
      b. Membuat laporan pemasukkan dan pengeluaran keuangan distro.
      c. Memberikan laporan keuangan ke pemilik distro
2. Distro Shut Up
   1) Pemilik Distro

a. Bertanggung jawab atas keseluruhan yang ada di distro Shut Up.
b. Bertugas untuk salah satu tugas dari kasir dan petugas atau dua-duanya apabila pemilik distro dan atau salah satu pegawai tidak hadir
c. Bertugas menerima laporan stok barang dari petugas distro.
d. Menerima laporan pemasukkan dan pengeluaran keuangan dari kasir distro.

2) Petugas Distro
a. Bertugas mencari ketersediaan stok barang untuk customer.
b. Menampilkan barang pada distro Shut Up.
c. Menyimpan stok barang yang tersedia di distro.
d. Memberikan laporan stok barang distro ke pemilik distro.

3) Kasir Distro
a. Bertugas membuat 2 struk pembelian untuk customer dan arsip distro.
b. Membuat laporan pemasukkan dan pengeluaran keuangan distro.
c. Memberikan laporan keuangan ke pemilik distro.

3. Distro Pandemonium

1) Pemilik Distro
a. Bertanggung jawab atas keseluruhan yang ada di distro Pandemonium.
b. Bertugas untuk salah satu tugas dari kasir dan petugas atau dua-duanya apabila pemilik distro dan atau salah satu pegawai tidak hadir
c. Bertugas menerima laporan stok barang dari petugas distro.
d. Menerima laporan pemasukkan dan pengeluaran keuangan dari kasir distro.

2) Petugas Distro
a. Bertugas mencari ketersediaan stok barang untuk customer.
b. Menampilkan barang pada distro Pandemonium.
c. Menyimpan stok barang yang tersedia di distro.
d. Memberikan laporan stok barang distro ke pemilik distro.

3) Kasir Distro
a. Bertugas membuat 2 struk pembelian untuk customer dan arsip distro.

b. Membuat laporan pemasukkan dan pengeluaran keuangan distro.
c. Memberikan laporan keuangan ke pemilik distro.

4. Distro Kandang Babi 31

1) Pemilik Distro
   a. Bertanggung jawab atas keseluruhan yang ada di distro Kandang Babi 31.
   b. Bertugas untuk salah satu tugas dari kasir dan petugas atau dua-duanya apabila pemilik distro dan atau salah satu pegawai tidak hadir
   c. Bertugas menerima laporan stok barang dari petugas distro.
   d. Menerima laporan pemasukkan dan pengeluaran keuangan dari kasir distro.
2) Petugas Distro
   a. Bertugas mencari ketersediaan stok barang untuk customer.
   b. Menampilkan barang pada distro Kandang Babi 31.
   c. Menyimpan stok barang yang tersedia di distro.
   d. Memberikan laporan stok barang distro ke pemilik distro.
3) Kasir Distro
   a. Bertugas membuat 2 struk pembelian untuk customer dan arsip distro.
   b. Membuat laporan pemasukkan dan pengeluaran keuangan distro.
   c. Memberikan laporan keuangan ke pemilik distro.

2.2 Landasan Teori

Landasan teori adalah kumpulan dari teori-teori yang dijadikan sebagai sumber referensi dalam pembangunan aplikasi ini yang dikutip sebagai sumber buku dan jurnal-jurnal elektronik. Landasan teori dimanfaatkan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan. Selain itu landasan teori juga bermanfaat untuk memberikan gambaran umum tentang latar penelitian dan sebagai bahan pembahasan hasil penelitian. Peneliti mengutip beberapa teori yang berhubungan dengan variabel-variabel penelitian. Dan teori-teori ini merupakan landasan dalam penelitian.

2.2.1 Aplikasi

Aplikasi berasal dari kata *application*[3] yaitu bentuk benda dari kata kerja to apply yang dalam bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

2.2.1.1 Jenis Aplikasi Komputer

Berdasarkan jenisnya aplikasi komputer dapat dikategorikan menjadi beberapa jenis aplikasi komputer yaitu[3] :

1. *Enterprise*

   Digunakan untuk organisasi yang cukup besar dengan maksud menghubungkan aliran data dan kebutuhan informasi antar bagian.
2. *Enterprise – Support*

   Sebagai aplikasi pendukung dari Enterprise.
3. *Individual Worker*

   Sebagai aplikasi yang biasa digunakan untuk mengolah/edit data oleh tiap individu.
4. Aplikasi Akses Konten

   Aplikasi yang digunakan oleh individu untuk mengakses konten tanpa kemampuan untuk mengolah/mengedit datanya melainkan hanya melakukan kustomisasi terbatas.
5. Aplikasi pendidikan

   Biasanya berbentuk simulasi dan mengandung konten yang spesifik untuk pembelajaran.
6. Aplikasi Simulasi

   Biasa digunakan untuk melakukan simulasi penelitian, pengembangan dll.
7. Aplikasi Pengembangan Media

   Berfungsi untuk mengolah/mengembangkan media biasanya untuk kepentingan komersial, hiburan dan pendidikan.
8. Aplikasi Mekanika dan Produk

   Dibuat sebagai pelaksana/pengolah data yang spesifik untuk kebutuhan tertentu.

2.2.1.2 Klasifikasi Aplikasi Yang Digunakan

Aplikasi dapat digolongkan menjadi beberapa klasifikasi kelas perangkat lunak, namun klasifikasi yang digunakan dalam aplikasi ini adalah Perangkat Lunak Infrastruktur Perusahaan yaitu, Sistem Manajemen Basis Data.

Sistem manajemen basis data atau *Database Management System*, (DBMS), atau kadang disingkat SMBD, adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna[4]. Sebagai contoh DBMS adalah *oracle*, SQL *server* 200/2003, MS *Access*, Mysql dan sebagainya.

2.2.2 Internet

Internet adalah jaringan komunikasi global yang terbuka dan menghubungkan ribuan jaringan komputer, melalui sambungan telepon umum maupun pribadi (pemerintah maupun swasta). Secara individual, jaringan komponennya dikelola oleh agen-agen pemerintah, universitas, organisasi komersial, maupun sukarelawan.

Internet muncul dari jaringan jarak jauh yang dikembangkan oleh ARPA net(*US Government's Advanced Research Projects Agency Network* Atau Jaringan Agen Proyek Riset Lanjutan dari Pemerintahan Amerika Serikat) di akhir tahun 60-an. Sambungan telepon cepat yang bervolume tinggi ternyata dapat diandalkan, sehingga jaringan ini kemudian diperluas selama 10 tahun berikutnya untuk menghubungkan 200 komputer di lembaga-lembaga riset maupun militer di AS dan seluruh dunia. Ini secara jelas membuktikan bahwa hubungan antar jaringan *(internetworking*) ternyata praktis dan sangat bermanfaat. Beberapa universitas di AS kemudian mengikutinya dengan membangun sistem mereka sendiri. Dalam pertengahan tahun 80-an, mereka bergabung dengan bagian riset dari ARPAnet, sehingga terbentuk Internet. Hal yang penting untuk dicatat disini adalah bahwa Internet tidaklah dibangun untuk usaha komersial. Sehingga masih ada ketentuan *Appropriate Use* (Penggunaan Yang Semestinya*)*, yang membatasi pemanfaatan Internet untuk kepentingan mencari keuntungan.

Tetapi sekarang, sebagian terbesar dari universitas-universitas di dunia telah tersambung, baik secara langsung maupun tidak langsung. Sementara kalangan bisnis banyak yang bergabung. Sebagian karena ingin memanfaatkan komunikasi internasionalnya yang efisien dan murah, sebagian lagi ingin mengiklankan barangnya dan atau jasa pelayanannya. Namun sebagian lainnya ingin mendapatkan akses ke data online yang luar biasa banyaknya itu. Sejumlah partai politik, kelompok-kelompok yang berpengaruh, maupun badan-badan amal, juga menggunakannya untuk menghubungkan para anggota mereka dan untuk menyampaikan pesan-pesan mereka. Organisasi-organisasi ini membawa jutaan orang masuk ke Internet. Meskipun demikian, ada pula jutaan individu yang menghubungkan komputer-komputer di rumah mereka melalui salah satu dari para *provider* atau perusahaan yang memberikan layanan internet ini. Di Inggris saja, lebih dari 100 ribu host komputer bergabung ke Internet. Mengenai individu-individu ini, *CompuServe*, yang merupakan *provider* yang paling besar menyatakan bahwa anggota mereka di Inggris telah mencapai 60.000 orang dengan 1.000 user (pemakai) baru yang bergabung setiap harinya.

2.2.3 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Selain itu pengertian Basis Data menurut Fathansya adalah sekumpulan data *persistence* yang saling terkait, menggambarkan suatu organisasi (*Enterprise*)[5].

Basis data digunakan karena memiliki keuntungan sebagai berikut :

1. Mengurangi redudansi
2. Data dapat di share antar aplikasi
3. Dapat melakukan standarisasi data
4. Batasan security dapat diterapkan
5. Mengelola integrasi data (akurasinya terjamin)

6. Indepedensi data (objek DBS), basis data dapat berkembang tanpa mempengaruhi aplikasi yang telah ada sebelumnya.

Secara definitif, basisdata merupakan suatu objek terstruktur tersebut terdiri atas data dan metadata. Data pada basis data merupakan informasi deskriptif yang benar-benar tersimpan, misalnya Nama atau Alamat. Sedangkan metadata merupakan bagian yang menjelaskan tentang struktur data tersebut dalam basis data, misalnya field untuk Nama dan Alamat, panjang field, atau tipe data untuk masing-masing field. Untuk menentukkan struktur suatu basis data digunakan pemodelan basis data. Pemodelan basis data ada banyak macamnya. Beberapa diantaranya yaitu :

1. *File System*
2. *Hirarchical Database Model*
3. *Network Database Model*
4. *Relational Database Model*

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa file teks ataupun *Database Management System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi [5] :

1. Memasukkan, menyimpan, dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan.

2.2.4 *Database Management System* (DBMS)

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan
4. Mempu menangani *backup* data

Karena pentingnya data bagi suatu organisasi atau perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang

mereka miliki. Pengelola DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut sebagai DBA (*Database Administrator*).

DBMS sudah mulai berkembang sejak tahun 1960-an. Kemudian sekitar tahun 1970-an mulai berkembang teknologi *Relational DBMS* yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana relasional model dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan *open source* saat ini berbasis *Relational* DBMS atau RDBMS.

Pada tahun 2980-an mulai berkembang *Object Oriented* DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemograman berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek. Saat ini OODBMS juga cukup berkembang, namun belum dapat menggeser kepopuleran RDBMS.

Berikut ini adalah 4 macam DBMS versi komersial yang paling banyak digunakan di dunia saat ini :

1. *Oracle*
2. *Microsoft SQL Server*
3. *IBM DB2*
4. *Microsoft Accsess*

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut :

a. *MySQL*
b. *PostgreSQL*
c. *Firebird*
d. *SQLite*

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS.

2.2.5 *Object Oriented Programing*

OOP adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan

operasi yang diberlakukan terhadapnya. Metode OOP didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode ini meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemograman berorientasi objek, dan pengujian berorientasi objek [6]. Ketika sebuah program berjalan, objek individu biasanya tidak dapat berdiri sendiri.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut.
2. Kecepatan pengembangan

   Karen sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pasa saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistenso pada saat pengembangannya, perangkat lunak dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan [6].

   Konsep dasar dari sebuah OOP adalah sebagai berikut :

   1. Objek, yaitu benda di dunia nyata yang bisa dibedakan antara satu dengan yang lainnya. Objek mempunyai identitas, property, dan tingkah laku.
   2. Kelas,yaitu sekumpulan objek yang memiliki kemiripan dalam hal properti, atribut, *behaviour*, dan semantik.

3. Abstraksi, yaitu mekanisme yang memungkinkan situasi dunia nyata dapat diwakili menggunakan model yang disederhanakan. Orientasi objek abstrak dunia nyata didasarkan pada objek dan interaksi antar objek lainnya.
4. Enkapsulasi dikenal sebagai informasi yang tersembunyi juga berperan pada pembungkusan properti dan *behaviour*. Enkapsulasi ini memastikan pengguna sebuah objek tidak dapat mengganti keadaan dari sebuah objek dengan cara yang tidak layak.
5. Polimorfisme, yaitu karakteristik sistem orientasi objek. Ketika warisan digunakan untuk memperpanjang *class* umum ke *class* yang lebih khusus, biasanya akan mencakup beberapa perilaku dari *class* umum. Perilaku dari *class* khusus ini akan berbeda dengan *class* umum, tetapi nama yang digunakan untuk mendefinikan perilaku tersebut akan menjadi sama.

Inheritas, yaitu mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

2.2.6 *Unified Modeling Language* (UML)

UML adalah salah satu bentuk *language* atau bahasa untuk menggambarkan *object oriented analysis and design.* Sebagian besar metode terdiri bahasa dan proses. UML disebut bahasa pemodelan bukan metode. Bahasa pemodelan adalah notasi (terutama grafis) bahwa metode digunakan untuk mengekspresikan desain.



Gambar 2.2 Logo UML

2.2.7 Sejarah UML

Bahasa pemograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemograman ini kurang berkembang dan dikembangkan lebih lanjut, namun kemunculannya menjadi titik besar bagi developer pengembang bahasa pemograman berorientasi objek selanjutnya.

Karena banyaknya metodologi yang berkembang pesat pada saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuatlah bahasa gabungan dari beberapa konsep seperti *Object Modelling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivan Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)* [6].

2.2.8 Diagram UML

Menggunakan berbagai macam diagram dengan fungsi masing-masing untuk menggambarkan setiap proses dari sistem berorientasi objek. Berikut merupakan beberapa diagram UML diantaranya:

A. *Use Case Diagram*

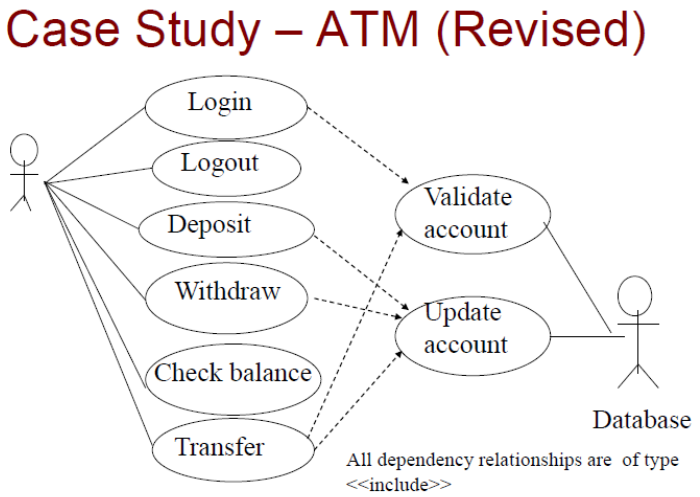
Use Case atau diagram *use case* merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (*behavior)* dari sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibuat diluar sistem yang akan dibuat itu sendiri, jadi

walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antarunit atau aktor.

Contoh dari *use case* diagram dapat dilihat pada gambar 2.3 berikut :



Gambar 2.3 Contoh dari *Use Case Diagram*

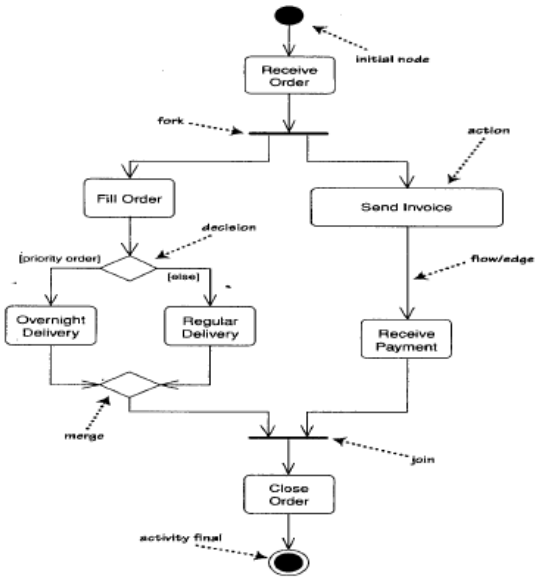
B. *Activity Diagram*

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

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

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

Contoh dari *activity diagram* dapat dilihat pada gambar 2.4 berikut :



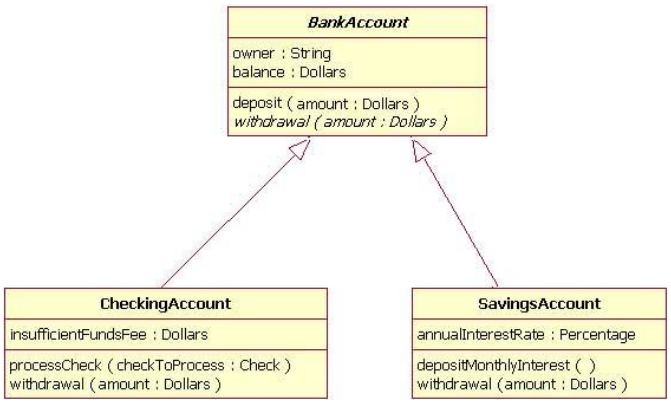
Gambar 2.4 Contoh dari *Activity Diagram*

C. *Class Diagram*

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

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Contoh dari *class diagram* dapat dilihat pada gambar 2.5 berikut :



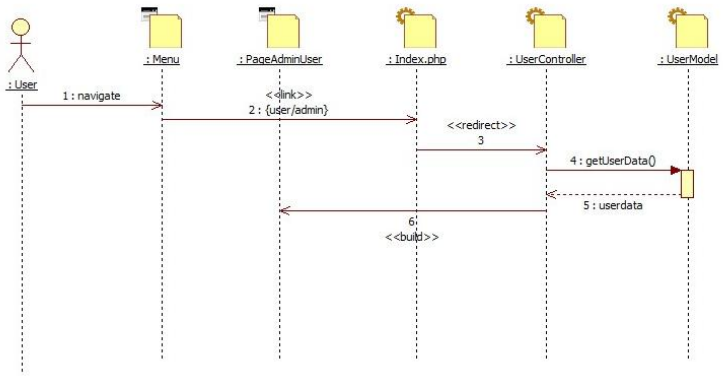
Gambar 2.5 Contoh dari *Class Diagram*

D. *Sequence Diagram*

Sequence diagram atau diagram sekuen adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang dinstansiasi menjadi objek itu.

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

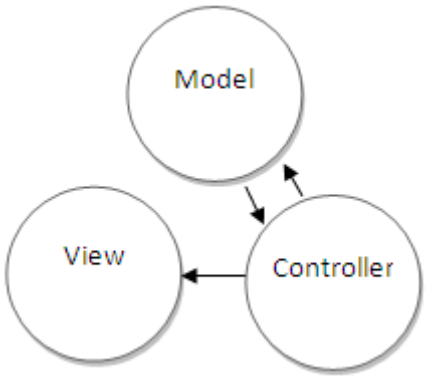
Contoh dari *sequence diagram* dapat dilihat pada gambar 2.6 berikut :



Gambar 2.6 Contoh dari *Sequence Diagram*

E. Model View Controller

Teknik pemrograman MVC (Model View Controller) merupakan teknik pemrograman yang populer saat ini, yang mengharapkan pemrogram secara disiplin untuk membagi program menjadi tiga bagian; model, view, dan controller, seperti gambar 2.7 berikut[11] :



Gambar 2.7 Arsitektur MVC [11]

1. Model

   Objek model adalah bagian dari aplikasi yang mengimplementasi logika untuk domain data aplikasi. Umumnya, objek model digunakan untuk mengambil data dari database atau menyimpan data ke dalam database.

2. View

   View adalah komponen yang menampilkan antarmuka untuk pengguna (*User* interface, UI) aplikasi. Antarmuka ini dibuat berdasarkan data dari model.

3. Controller

   Controller merupakan komponen yang digunakan untuk menangani interaksi pengguna, bekerja dengan model, dan memilih view mana yang digunakan untuk merender data.

Dalam aplikasi MVC, view hanya menampilkan informasi, controller menangani dan menanggapi masukan dan interaksi dengan pengguna. Dengan pola MVC, kita membuat aplikasi dengan memisahkan berbagai aspek dari aplikasi (logika input, bisnis, dan antarmuka), ada bagian penampil data (View), penyedia atau pemroses data (model), dan pengendali program (controller).

Dengan pemisahan ini, maka pembuatan satu program dapat dibagi-bagi kepada beberapa orang. Setiap bagian atau komponen yang dibuat secara terpisah diharapkan dapat digunakan tidak hanya oleh satu program saja, tetapi oleh banyak program, terutama bagian model dan view. Teknik pemrograman MVC ini

diimplementasi tidak hanya dalam PHP, tetapi suatu teknik pemrograman yang sudah dikenal dan digunakan hampir pada semua bahasa pemrograman yang ada.

2.3 Perangkat Lunak Pendukung

2.3.1 PHP5 (*Personal Home Page*)

PHP5 *(Personal Home Page)* merupakan secara umum dikenal sebagai bahasa pemograman script script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML. Dikenal juga sebagai bahasa pemprograman server side.

Dengan menggunakan PHP5 maka *maintence* suatu situs web menjadi lebih mudah, proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan script PHP5.

PHP5/FI merupakan nama awal dari PHP5. PHP5 – *Personal Home Page,* FI adalah *Form Interface*. Dibuat pertama kali oleh Rasmus Lerdoff. PHP5 awalnya merupakan program CGI yang dikhususkan untuk menerima *input* melalui form yang ditampilkan dalam browser web. *Software* ini disebarkan dan dilesensikan sebagai perangkat lunak *Open Source*.

Integrasi PHP5 dengan *server* web dilakukan dengan teknik CGI, FastCGI, dan modul *server* web. Teknik CGI dan FastCGI memisahkan antara *server* web dan PHP5, sedangkan modul *server* web menjadi PHP5 sebagai bagian dari *server* web.

PHP5 adalah kependekan dari *HyperText Preprocessor* merupakan bahasa utama script *server-side* yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop. Setiap program PHP5 disebut dengan *script*. Script berupa *file* teks, yang dapat dibuat dengan menggunakan program editor *file* teks biasa seperti *notepad*, *edit*, vi (dalam lingkungan Unix/Linux), ataupun lainnya.

Editor teks yang digunakan sebaiknya editor teks yang memungkinkan membuat program PHP5 lebih mudah. Penulis sendiri menggunakan Notepad++, editor teks *freeware* yang bagus untuk digunakan.

Script PHP5 diawalai dengan tag <?PHP5 kemudian diakhiri dengan tag ?> .script PHP5 merupakan script yang digunakan untuk menghasilkan halaman-halaman web. Cara penulisannya dibedakan menjadi *embedded* dan *non embedded script* yang akan dijelaskan sebagai berikut :

1. *Embedded Script*

   Berikut adalah contoh dokumen HTML yang akan dihasilkan dengan menggunakan program atau script PHP5 dalam *embedded script :*

   Beri nama *file :* sc_embedded.PHP

```
<html>
<head>
<title>Contoh Embadded</title>
</head>
<body>
<?PHP
echo “Hello World -Unleashed !”;
?>
</body>
</html>
```

Script diatas menunjukkan contoh script PHP5 sederhana, yang disebut dengan *embedded script*. *Script* PHP5 disisipkan diantara tag-tag HTML. *Scrip*t PHP5 digunakan apabila isi dari suatu dokumen HTML diinginkan dari hasil eksekusi suatu *script* PHP5, selama informasi masih tidak membutuhkan program maka pemogram umumnya tidak akan menggunakan program.

2. *Non Embadded Script*

   Script PHP5 pada cara ini digunakan sebagai murni pembuatan program dengan PHP5, tag HTML yang dihasilkan untuk membuat dokumen merupakan bagian dari *script* PHP5.

   Beri nama *script* sc_nonembedded.PHP :

```
<?PHP
echo “<html>”;
```

```
echo “<head>”;
echo “      <title>”;
echo “Contoh Non Embadded – Pure On the Fly”;
echo ” </title>”;
echo ”</head>”;
echo ”<body>”;
echo “<p> Hello World –Unleashed !</p>”;
echo “ </body>
echo “</html>
?>
```

Source dokumen HTML dari tampilan sc_nonembedded.PHP sebagai berikut:

```
<html><head><title>          Contoh Non Embedded Pure On the Fly
</title></head><body><p> Teks Dokumen yang dihasilkan dengan
Menggunakan script PHP </p></body></html>
```

Tampak tag HTML dalam dokumen tersebut tidak beraturan ditampilkan. Tidak menjadi masalah, yang penting adalah *browser* web dapat menampilkannya, karena dokumen dan tag HTML ini murni dihasilkan oleh *script* PHP5.

2.3.2 MySQL (*My Structure Query Language*)

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan standar SQl *(Structured Query Language).* MySQL memiliki dua bentuk lisensi, yaitu *FreeSoftware* dan *Shareware*. MySQL yang biasa kita gunakan adalah MySQL *FreeSoftware* yang berada di bawah Lisensi GNU/GPL (*Generic Public Lisence*).

MySQL merupakan sebuah *database server* yang free, artinya kita bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali diliris oleh seorang programmer *database* bernama Michael Widenius. Selain sebagai *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai server. Pada saat itu berarti program berposisi sebagai *Clien*t. Jadi MySQL adalah sebuah *database* yang dapat digunakan baik sebagai *Client* atau *Server*.

Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relational atau dalam bahasa basis data sering disebut dengan *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan bernama SQL. MySQL adalah sebuah program database, sedangkan SQL adalah bahasa perintah (*Query*) dalam program MySQL.

MyQSL adalah sebuah database *server*, dapat juga berperan sebagai *client* sehingga sering disebut database *client/server*, yang open source dengan kemampuan dapat berjalan dengan baik di OS (*Operating System*) manapun dengan Platform Windows maupun Linux. Selain itu database ini memiliki beberapa kelebihan dibanding database lain, diantaranya adalah :

1. MySQL sebagai *Database Management System* (DBMS)
2. MySQL sebagai *Relation Database Management System* (RDBMS)
3. MySQL adalah sebuah *Software database* yang *OpenSource*, artinya program ini bersifat free atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
4. MySQL merupakan sebuah *database server*, jadi dengan menggunakan database ini kita dapat menghubungkannya ke media internet sehingga cepat diakses dari jauh.
5. MySQL merupakan sebuah *database client*. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *Query* yang mengakses database pada *server*. Jadi MySQl dapat juga berperan sebagai *Client*.
6. MySQl mampu menerima *query* yang bertumpuk dalam satu permintaan

atau yang disebut *Multi-Threading*.

7. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *GigaByte* sekalipun.
8. MySQL didukung oleh *driver* ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa *visual* seperti *Delphi* maupun *Visual Basic.*
9. MySQL adalah *database* menggunkan enkripsi *password*. Jadi database ini cukup aman karena memiliki *password* untuk mengaksesnya.
10. MySQL merupakan *Server database* yang *multi user*, artinya *database* ini tidak hanya digunakan oleh sepihak orang akan tetapi merupakan database yang dapat digunakan oleh banyak pengguna.
11. MySQL dapat menciptakan lebih dari 16 kunci pertabel, dan dalam satu kunci memungkinkan berisi belasan *field*.
12. MySQL mendukung *field* yang menjadikan sebagai kunci primer dan kunci *Unique*.
13. MySQL didukung oleh sebuah komponen C dan perl API, sehingga *Database* MySQL dapat diakses melalui sebuah program aplikasi yang berada dibawah protokol intenet berupa web. Biasanya aplikasi yang sering digunakan adalah PHP5.
14. MySQL memiliki kecepatan dalam pembuatan tabel ataupun peng-*update*-an tabel.
15. MySQL menggunakan suatu bahasa permintaan standar yang bernama SQL (*Structured Query Language*) yaitu sebuah bahasa permintaan yang distandarkan pada beberapa database *serve*r seperti *Oracle, PostGreSQL* dll.

SQL (*Structured Query Language*) merupakan bahasa ANSI (*American Nasional Standart Input*) yang digunakan untuk melakukan *query* data pada database. Semua pengoperasian data dapat dikerjakan secara mudah dengan menggunakan bahasa ini, terutama dalam pemasukan dan seleksi.

Bahasa SQL memiliki struktur yang mudah dipahami, karena menggunkan perintah-perintah SQL pada *database* MySQL seperti :

1. Memasukkan atau menambahkan *record* baru kedalam database
2. Mengeksekusi *query database*
3. Mengambil data dari *database*
4. Mengubah record pada *database*
5. Menghapus record pada *database*

Berikut ini akan dijelaskan contoh-contoh pembuatan sebuah *database* menggunakan perintah-perintah SQL :

a. Create

```
CREATE TABLE siswa(
    id int (7)
    nama CHAR(20)

      alamat char(30)
);
```

Perintah diatas akan membuat sebuah tabel dengan nama siswa, yang berisi kolom id dengan tipe *integer* dengan panjang 7 karakter, nama tipe char dengan panjang 20 karakter dan alamat tipe char dengan panjang 30 karakter.

b. Select

```
SELECT * FROM siswa;
```

Perintah *select* akan menampilkan semua data yang ada di tabel siswa.

c. Insert

```
INSERT INTO siswa
 VALUES (1234567,'Gita','Bandung');
```

Perintah tersebut digunakan untuk mengisi tabel siswa dengan banyak value seperti id, nama serta alamat.

d. Update

```
UPDATE siswa
    SET nama='Gita'
     WHERE id=1234567;
```

Perintah *update* dipakai untuk merubah atau mengganti data yang ada pada *database.*

e. Delete

```
DELETE FROM user
    WHERE user = ‘Gita’;
```

Perintah ini dilakukan untuk menghapus salah satu value yang ada di tabel.

2.3.3 HTML

Hypertext Markup Language merupakan kepanjangan dari kata HTML. HTML adalah *script* dimana kita bisa menampilkan informasi dan daya kreasi kita lewat internet. HTML juga merupakan file teks murni yang dapat dibuat dengan editor teks sembarang yaitu yang dikenal sebagai web *page* atau dokumen yang disajikan dalam web browser. Dokumen ini umumnya berisi informasi atau *interface* aplikasi didalam internet.

HTML sendiri adalah suatu dokumen teks biasa yang mudah dimengerti dibandingkan bahasa pemograman lainnya, dan karena bentuknya itu maka HTML dapat dibaca oleh berbagai *platform* seperti : *windows, Linux, Macintosh.* Kata *Markup Language* pada HTML menunjujjan fasilitas yang berupa tanda tertentu dalam *script* HTML sehingga kita bisa mengatur judul, garis, tabel, gambar dan lain-lain dengan perintah yang telah ditentukkan dalam elemen HTML. Simbol penandaan yang digunakan dalam HTML ditandai dengan tanda awal < dan diakhiri dengan tanda > yang disebut dengan tag. Contohnya :

```
<html>
<body>
<p>Hallo Ini adalah contoh penulisan HTML </p>
</body>
</html>
```

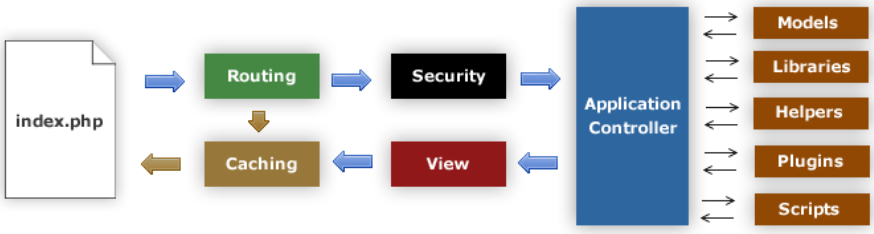
2.3.4 *CodeIgniter*

CodeIgniter (CI) adalah *framework* pengembangan aplikasi (Application Development Framework) dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis[11]. Pemrogram tidak perlu membuat program dari awal (*from scracth*), karena CI menyediakan sekumpulan librari yang banyak yang diperlukan untuk menyelesaikan pekerjaan yang umum, dengan menggunakan antarmuka dan struktur logika yang sederhana untuk mengkakses librarynya. Pemrogram dapat memfokuskan diri pada kode yang harus dibuat untuk menyelesaikan suatu pekerjaan.

Framefork CI merupakan framework yang memiliki dokumentasi yang jelas dan lengkap, yang memudahkan pengembang untuk mempelajari dengan mudah. Pendekatan dari CI sangatlah mudah, dari membuat sekadar tulisan sampai dengan yang kompleks dapat didekati dengan mudah. Tidak seperti framework yang lain, untuk mendapatkan tulisan Hello World di browser saja, kita harus menggunakan beberapa tahap. CI cukup dengan satu file dan satu prosedur atau method.

2.3.5 Pemrograman MVC dengan *CodeIgniter*

CI adalah satu framework pemrograman PHP yang mengimplementasikan teknik pemrograman MVC. Berikut adalah arsitektur bagaimana CI mengimplementasikan MVC[11].



Gambar 2.8 Arsitektur *CodeIgniter*[11]

Index.php berlaku sebagai controller di depan, yang menginisialisasi sumber daya yang dibutuhkan untuk menjalankan *CodeIgniter*

1. Router akan memeriksa request HTTP untuk menentukan apa yang harus dilakukan dengan request tersebut
2. Jika ada file cache, maka file cache langsung dikirimkan kepada browser, dengan memotong eksekusi sistem normal
3. Keamanan. Sebelum suatu aplikasi controller dipanggil, request HTTP dan data yang dikirimkan oleh pengguna disaring (filter) terlebih dahulu untuk keamanan
4. Controller akan memuat model, libari inti, helper, dan sumberdaya lainnya yang diperlukan untuk memroses request tersebut
5. View terakhir direnderkemudian dikirimkan kepada browser untuk dilihat. Jika caching didefinisikan, maka view akan dicache terlebih dahulu, setelah itu baru dikirimkan kepada browser.

Codeigniter (CI) bisa digunakan untuk belajar melakukan pemrograman dengan teknik pemrograman yang menggunakan konsep MVC (Model View Controller). CI mengijinkan pemrogram untuk bekerja penuh di dalam controller, sambil memahami konsep MVC dengan benar.

2.4 Pengujian *Blackbox*

Pengujian yang dilakukan untuk antarmuka perangkat lunak, pengujian ini dilakukan untuk memperlihatkan bahwa fungsi-fungsi bekerja dengan baik dalam arti masukan yang diterima dengan benar dan keluaran yang dihasilkan benar-benar tepat, pengintegrasian dari eksternal data berjalan dengan baik.

Metode pengujian *blackbox* memfokuskan pada *requirement* fungsi dari perangkat lunak, pengujian *blackbox* dilakukan pada tahap akhir dari pengujian perangkat lunak. Proses yang terdapat dalam proses pengujian *blackbox* antara lain sebagai berikut:

1. Pembagian kelas data untuk pengujian setiap kasus yang muncul pada pengujian *whitebox*.
2. Analisis batasan nilai yang berlaku untuk setiap data

2.5 Kuesioner

Kuesioner merupakan teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya. Kuesioner merupakan teknik pengumpulan data yang efisien bila peneliti tahu dengan pasti variabel yang akan diukur dan tahu apa yang bisa diharapkan dari responden. Selain itu, kuisioner juga cocok digunakan apabila jumlah responden cukup besar dan tersebar di wilayah yang luas. Kuesioner dapat berupa pertanyaan atau pernyataan tertutup atau terbuka, dapat diberikan kepada responden secara langsung atau dikirim melalui pos atau internet.

Bila penelitian dilakukan pada lingkup yang tidak terlalu luas, sehingga kuesioner dapat diantarkan langsung dalam waktu tidak terlalu lama, maka pengiriman angket pada responden tidak perlu melalui pos. Dengan adanya kontak langsung antara peneliti dengan responden akan menciptakan suatu kondisi yang cukup baik, sehingga responden dengan sukarela akan memberikan data objektif dan cepat.

Suharsimi Arikunto (2004) mengemukakan beberapa instrumen dalam pengumpulan data yaitu :*alat bantu yang dipilih dan digunakan oleh peneliti dalam keegiatan tersebut menjadi sistematos dan lebih mudah dilakukan*[10].

1. Prinsip Penulisan Angket

   Prinsip ini menyangkut beberapa faktor yaitu : isi dan tujuan pertanyaan, bahasa yang digunakan mudah, pertanyaan tertutup terbuka-negatif positif, pertanyaan tidak mendua, tidak menanyakan hal-hal yang sudah lupa, pertanyaan tidak mengarahkan, panjang pertanyaan, dan urutan pertanyaan.

   a. Isi dan tujuan Pertanyaan

      Yang dimaksud disini adalah apakah isi pertanyaan tersebut merupakan bentuk pengukuran atau bukan. Kalau berbentuk pengukuran, maka dalam membuat pertanyaan harus teliti, setiap pertanyaan harus skala pengukuran dan jumlah itemnya mencukupi untuk mengukur variabel yang diteliti

   b. Bahasa yang digunakan

      Bahasa yang digunakan dalam penulisan kuesioner (angket) harus disesuaikan dengan kemampuan berbahasa responden. Kalau sekiranya

responden tidak dapat berbahasa Indonesia, maka angket jangan disusun dengan bahasa Indonesia. Jadi bahasa yang digunakan dalam angket harus memperhatikan jenjang pendidikan responden, keadaan sosial budaya, dan "*frame of reference*" dari responden.

c. Tipe dan Bentuk Pertanyaan

Tipe pertanyaan dalam angket dapat terbuka atau tertutup, (kalau dalam wawancara : terstruktur dan tidak terstruktur), dan bentuknya dapat menggunakan kalimat positif atau negatif.

Pertanyakan terbuka, adalah pertanyaan yang mengharapkan responden untuk menuliskan jawabannya berbentuk uraian tentang sesuatu hal. Contohnya : bagaimanakah tanggapan anda terhadap iklan-iklan di TV saat ini ? sebaliknya pertanyaan tertutup, adalah pertanyaan yang mengharapkan jawaban singkat atau mengharapkan responden untuk memilih salah satu alternatif jawaban dari setiap pertanyaan yang telah tersedia. Setiap pertanyaan angket yang mengharapkan jawaban berbentuk *data nominal, ordinal, interval, dan ratio, adalah bentuk pertanyaan tertutup.*

Pertanyaan tertutup akan membantu responden untuk menjawab dengan cepat, dan juga memudahkan peneliti dalam melakukan analisis data terhadap seluruh angket yang telah terkumpul. Pertanyaan atau pernyataan dalam angket perlu dibuat kalimat positif dan negatif agar responden dalam memberikan jawaban setiap pertanyaan lebih serius, dan tidak mekanistis.

d. Pertanyaan tidak mendua

Setiap pertanyaan dalam angket jangan mendua (*double-barreled*) sehingga menyulitkan responden untuk memberikan jawaban.

e. Tidak menanyakan yang sudah lupa

Setiap pertanyaan dalam instrumen angket, sebaiknya juga tidak menanyakan hal-hal yang sekiranya responden sudah lupa, atau pertanyaan yang memerlukan jawaban dengan berfikir berat.

f. Pertanyaan tidak menggiring

Pertanyaan dalam angket sebaiknya juga tidak menggiring ke jawaban yang baik saja atau ke yang jelek saja.

g. Panjang Pertanyaan

Pertanyaan dalam angket sebaiknya tidak terlalu panjang, sehingga akan membuat jenuh responden dalam mengisi. Bila jumlah variabel banyak, sehingga memerlukan instrument yang banyak, maka instrumen tersebut dibuat bervariasi dalam penampilan, model skala pengukuran yang digunakan, dan cara mengisinya. Disarankan empirik jumlah pertanyaan yang memadai adalah antara 20 s/d 30 pertanyaan.

h. Urutan Pertanyaan

Urutan pertanyaan dalam angket, dimulai dari yang umum menuju ke hal yang spesifik, atau dari yang mudak menuju ke hal yang sulit, atau diacak. Hal ini perlu dipertimbangkan karena secara psikologis akan mempengaruhi semangat responden untuk menjawab. Kalau pada awalnya sudah diberikan pertanyaan yang sulit, atau yang spesifik, maka responden akan patah semangat untuk mengisi angket yang telah mereka terima. Urutan pertanyaan yang diacak perlu dibuat bila tingkat kematangan responden terhadap masalah yang ditanyakan sudah tinggi.

2. Prinsip Pengukuran

Angket yang diberikan kepada responden adalah merupakan instrumen penelitian, yang digunakan untuk mengukur variabel yang akan diteliti. Oleh karena itu instrumen angket tersebut harus dapat digunakan untuk mendapatkan data yang valid dan reliabel tentang variabel yang diukur. Supaya diperoleh data penelitian yang valid dan reliabel, maka sebelum instrumen angket tersebut diberikan pada responden maka perlu diuji validitas dan realibilitasnya terlebih dulu. Instrumen yang tidak valid dan reliabel bila digunakan untuk mengumpulkan data, akan menghasilkan data yang tidak valid dan reliabel pula.

3. Penampilan Fisik Angket

Penampilan fisik angket sebagai alat pengumpul data akan mempengaruhi respon atau keseriusan responden dalam mengisi angket. Angket yang dibuat dikertas buram, akan mendapatkan respon yang kurang baik bagi responden, bila dibandingkan dengan angket yang dicetak dalam kertas yang bagus dan

berwarna. Tetapi angket yang dicetak dikertas yang bagus dan berwarna akan menjadi lebih mahal.

2.5.1 *Skala Likert*

Skala likert digunakan untuk mengukur sikap, pendapat, dan presepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian. Dengan *skala Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan.

Jawaban setiap item instrumen yang menggunakan skala likert mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata antara lain[10] :

1. Sangat Setuju	a. Selalu
2. Setuju	b. Sering
3. Ragu-ragu	c. Kadang-kadang
4. Tidak setuju	d. Tidak pernah
5. Sangat tidak setuju	
a. Sangat positif	a. Sangat baik
b. Positif	b. Baik
c. Negatif	c. Tidak baik
d. Sangat negatif	d. Sangat tidak baik

Untuk keperluan analisis kuantitatif, maka jawaban itu dapat diberi skor, misalnya :

1. Setuju/selalu/sangat positif diberi skor 5
2. Setuju/sering/positif diberi skor 4
3. Ragu-ragu/kadang-kadang/netral diberi skor 3
4. Tidak setuju/hampir tidak pernah/negatif diberi skor 2
5. Sangat tidak setuju/tidak pernah diberi skor 1

Instrumen penelitian yang menggunakan *Skala Likert* dapat dibuat dalam bentuk checklist ataupun pilihan ganda.

a. Contoh Bentuk *Checklist*

Berilah jawaban pertanyaan berikut sesuai dengan pendapat anda, dengan cara memberi tanda (√) pada kolom yang tersedia.

Tabel 2.1 Contoh Tabel *Checklist*

No	**Pertanyaan**	**Jawaban**				
		SS	**S**	**RG**	**TD**	**STS**
1	Prosedur kerja yang baru itu akan segera diterapkan di perusahaan anda		√			
2	.....					

SS	= Sangat Setuju	diberi skor	5
S	= Setuju	diberi skor	4
RG	= Ragu-ragu	diberi skor	3
TS	= Tidak Setuju	diberi skor	2
STS	= Sangan Tidak setuju	diberi skor	1

Kemudian dengan teknik pengumpulan data angket, maka contoh yang diambil sebagai responden misalnya diberikan kepada 70 orang karyawan yang diambil secara random. Dari 70 orang pegawai setelah dilakukan analisis misalnya :

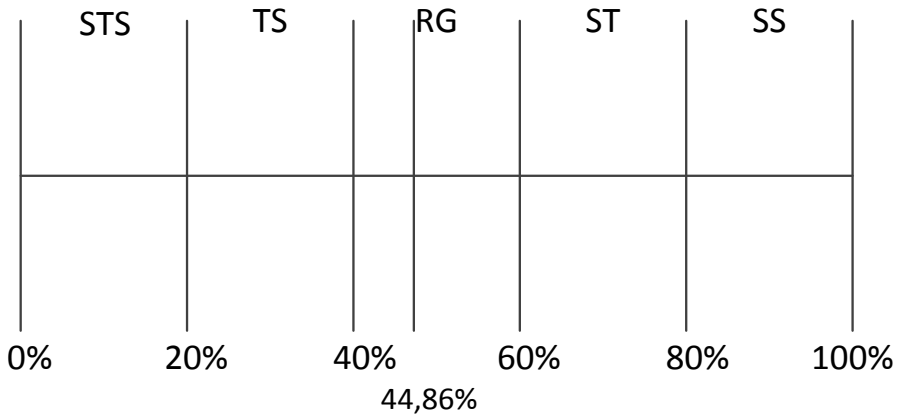
2 orang menjawab	SS	: 2 x 5 = 10
8 orang menjawab	S	: 8 x 4 = 32
15 orang menjawab	RG	: 15 x 3 = 45
25 orang menjawab	TS	: 25 x 2 = 50
20 orang menjawab	STS	: 20 x 1 = 20
		Jumlah : 157

Jumlah skor ideal (skor tertinggi) = 5 x 70 = 350 (SS)

Jumlah skor rendah = 1 x 70 = 70 (STS)

Berdasarkan data yang diperoleh dari 70 responden terletak pada daerah Netral dengan jumlah 157. Jadi diperoleh dari 70 responden, yaitu :

157 / 350 x 100% = 44,86%[10], tergolong ragu-ragu. Persentase dapat dilihat pada gambar 2.8 berikut :



Gambar 2.9 Persentase Skor

keterangan :

0% - 20%	Sangat Tidak Setuju
21% - 40%	Tidak Setuju
41% - 60%	Ragu-Ragu
61% - 80%	Setuju
81% - 100%	Sangat Setuju