BAB 2
TINJAUAN UMUM PERUSAHAAN

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah Provectus Cloth

Provectus Cloth adalah sebuah Distro atau Toko yang memproduksi dan menjual pakaian. Berlokasi di Jalan Dalem Kaum No.54 Plaza Parahyangan Lantai III Bandung, Selain menjual Provectus juga menerima jasa pembuatan perlengkapan pakaian yang lokasi pembuatanya berada di Cicaheum.

Provectus Cloth berdiri pertengan tahun 2013 tepatnya pada 21 April 2013, dimana seorang pendirinya lulusan Diploma 3 Universitas Widyatama Bernama Pandi Fauzi Ridwan A.md. Walaupun terbilang masih baru dan sangat muda untuk mengawali karir di dunia bisnis pakaian, Provectus telah memiliki nama tersendiri di para konsumennya dan mampu bersaing terhadap kompetitornya di dunia *Fashion.*

2.1.2 Visi dan Misi PROVECTUS CLOTH

Visi Provectus Cloth Menjadi Distro yang Profesional dan Inovatif dalam membuat model perkembangan pakaian terkini.

Misi Provectus Cloth adalah :

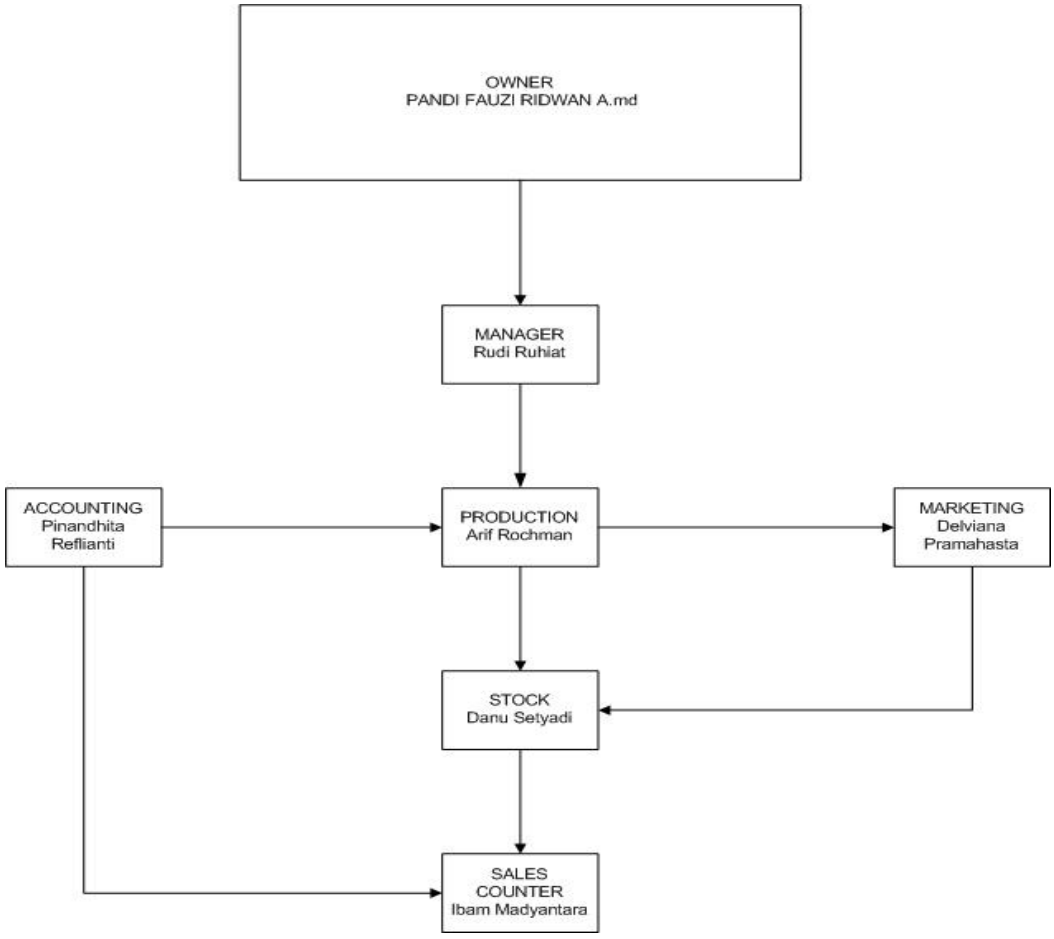
- Mengembangkan produk pakaian yang tepat guna.
- Mengedepankan profesionalisme dalam menghasilkan layanan yang berkualitas.
- Memberikan layanan yang maksimal bagi klien.

Mengembangkan inovasi Pakaian yang modis terbaik dan terkini dalam setiap produk pakaian yang di hasilkan.

2.1.3 Logo Provectus

Gambar 2 Logo Provectus Clot

2.1.4 Struktur Organisasi



Gambar 2.1 Struktur Oragnisasi

2.1.5 Deskripsi Pekerjaan

1. Pemilik (owner)

Tugas dan tanggung jawab pemilik ialah memberi arahan serta informasi penting berkaitan dengan Distro Provectus Cloth. Selain itu juga pemilik atau biasa kita sebut direktur ini bertugas mengambil keputusan penting untuk kemajuan Distro Provectus Cloth.

2. Manager

Tugas dan tanggung jawab manager adalah menjamin tercapainya hasil produksi dalam hal jumlah, kualitas dan waktu yang sesuai dengan rencana perusahaan dengan memanfaatkan sumber daya secara optimal memiliki tugas, wewenang dan tanggung jawabsebagai berikut :

a. Merencanakan dan mengatur jadwal produksi untuk semua jenis produk yang ditawarkan oleh Provectus Cloth agar tidak terjadi kelebihan atau kekurangan persediaan di gudang.
b. Mengatur pengalokasian sumber daya produksi seperti jam kerja mesin, jam kerjaoperator, pengiriman bahan baku yang berhubungan dengan proses produksi.
c. Melakukan pengawasan dan pengendalian produksi agar hasil produksi sesuai denganspesifikasi dan standart mutu yang telah ditetapkan.
d. Merencanakan perawatan mesin-mesin agar dapat beroperasi dengan lancar.

3. Produksi

Tugas dan tanggung jawab Produksi ialah membuat laporan produksi secara berkala mengenai pemakaian bahan baku dan bertanggung jawab terhadap kelancaran proses produksi mulai dari penerimaan bahan baku sampai proses produksi hingga menjadi produk akhir.

4. Akunting

Tugas dan tanggung jawab Akunting melaksanakan kegiatan administrasi keuangan Distro Provectus (jurnal entry sampai dengan arsip dokumen) untuk menjamin kelancaran, keakuratan dan ketertibanadministrasi keuangan Distro Provectus Cloth.

5. Marketing

Tugas dan tanggung jawab Marketing ialah melaksanakan kegiatan promosi di Distro Provectus Cloth.

6. Stok

Tugas dan tanggung jawab Stok ialah berhubungan dengan bagian produksi.

7. Sales Konter

Tugas Sales Konter ialah sebagai penjaga Distro Provectus yang berinteraksi dengan konsumen yang langsung datang ke tempat penjualan**.**

2.2 Landasan Teori

Kata system berasal dari bahasa Yunani (sustēma) dan Bahasa Latin (systēma). Banyak ahli yang mengungkapkan pendapat mereka mengenai pengertian sistem. Tetapi secara umum pengertian sistem yaitu "Sistem adalah sekumpulan unsur atau elemen yang saling berhubungan dan saling mempengaruhi satu sama lain yang melakukan kegiatan secara bersama-sama untuk mencapai tujuan yang sama pula."

2.2.1 Pengertian Informasi

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Suatu proses transformasi data menjadi suatu informasi yang didalamnya ada input,proses dan output.

2.2.2 Pengertian Sistem Informasi

Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan tertentu yaitu menyajikan informasi.

2.2.3 Pengertian Data

Data adalah sesuatu yang belum mempunyai arti bagi penerimanya dan masih memerlukan adanya suatu pengolahan. Data bisa berujut suatu keadaan, gambar, suara, huruf, angka, matematika, bahasa ataupun simbol-simbol lainnya

yang bisa kita gunakan sebagai bahan untuk melihat lingkungan, obyek, kejadian ataupun suatu konsep.

2.2.4 Pengertian Database

Database adalah kumpulan informasi yang disusun berdasarkan cara tertentu dan merupakan suatu kesatuan yang utuh, dengan sistem tersebut data yang terhimpun dalam suatu database dapat menghasilkan informasi yang berguna.

2.2.5 Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis sistem dan siklus pengembangan sistem, pendefinisian kebutuhan-kebutuhan fungsional dan persiapan untuk merancang bangun implementasi, menggambarkan suatu sistem yang akan dibentuk. Perancangan sistem dapat menggunakan pemodelan secara terstruktur menggunakan grafik atau diagram.

Langkah-langkah yang dilakukan pada tahap-tahap perancangan sistem ini adalah membuat usulan pemecahan masalah secara logika. Alat bantu yang digunakan adalah :

a. Struktur Data

   Struktur data merupakan hasil transformasi dari entitas-entitas pada diagram E-R.

b. Skema Relasi

c. Skema relasi (*relation schema*) adalah relasi yang memiliki nama, didefinisikan oleh himpunan pasangan attribute dan domainnya. Skema relasi terdiri dari :

   1) Nama Relasi harus unik
   2) Nama atribut relasi harus unik dan diasosiasikan dengan nama domain
   3) *Primary Key* (Kunci Primer/Utama) Atribut atau kombinasi atribut yang secara unik mengidentifikasi setiap baris dalam relasi

4) *Foreign Key* (Kunci Tamu) Atribut atau kombinasi atribut pada relasi yang berfungsi sebagai kunci primer pada relasi lain pada basisdata yang sama digunakan untuk melakukan referensi ke tupel pada relasi lain (relasi yang menjadikan kunci tersebut sebagai kunci primer)

d. *Use Case* Diagram

Use case diagram menggambarkan sejumlah eksternal actors dan hubunganya dengan *use case* yang diberikan oleh system. Dengan kata lain *use case* menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case* simbol namun dapat juga dilakukan dalam *activity* diagram. Dalam pemodelaan system dengan UML,actor adalah seseorang atau sesuatu yang berinteraksi dengan system yang sedang dikembangkan.

e. Sequence Diagram

Sequence Diagram menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga antara interaksi objek yang terjadi dalam titik tertentu dalam ekseskusi sistem.

Dalam UML, object dalam sequence digambarkan dengan segi empat yang berisi nama dari object yang digaris bawahi. Dalam sequence diagram setiap object hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antara object digambarkan dengan anak panah dari object yang mengirimkan pesan ke object yang menerima pesan.

f. Collaboration Diagram

Diagram Kolaborasi merupakan cara alternatif untuk menggambarkan skenario dari sistem. Diagram ini mengambarkan orientasi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek lainnya. Diagram kolaborasi berisi :

1) Objek digambarkan dengan dengan segi empat.

2) Hubungan antara objek digambarkan dengan garis penghubung.
3) Pesan yang digambarkan dengan text dan panah dari objek yang mengirim pean ke penerima pesan.

Diagram Sekuesial memberikan cara untuk melihat scenario dari system berdasarkan waktu (apa yang terjadi pertama kali, apa yang terjadi selanjutnya). Sedangkan Diagram kolaborasi cenderung untuk memberikan gambaran besar dari sekenario selama koloborasi disusun dari objek lainya.

g. Class Diagram

Kelas adalah sesuatu yang membungkus informasi atau atribut dan perilaku atau oprasi dalam dirinya. Diagram kelas menurut adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket – paket yang ada dalam sistem atau perangkat lunak yang sedang dikembangkan.

h. *Activity* Diagram

Activity Diagram menurut adalah salah satu cara untuk Diagram digunakan untuk memodelkan aspek dinamis dari system. *Activity* Diagram mirip dengan Flowchart memperlihatkan aliran kendali dari suatu aktifitas ke aktifitas lain.

i. Indikator *Multiplicity*

Walaupun *Multiplicity* ditentukan oleh class, *Multiplicity* menetukan banyaknya objek yang terlibat dalam relasi. *Multiplicity* menetukan banyaknya objek yang terhubung satu dengan yang lainya, Indikator *Multiplicity* terdapat pada masing – masing akhir garis relasi, baik pada asosiasi maupun agregasi. Beberapa contoh *Multiplicity* adalah:

1	Tepat satu
0..*	Nol atau lebih
1..*	Satu atau lebih
0..1	Nol atau satu

5..8	Range 5 s.d 8
4..6,9	Range 4 s.d 6 dan 9

j. Struktur Menu

Struktur menu adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer, sehingga saat menjalankan program user tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan.

2.2.6 Arsitektur Sistem

Arsitektur sistem merujuk pada konfigurasi sistem secara keseluruhan yang akan menjadi "tempat hidup" dari DBMS, basis datadan aplikasi yang memanfaatkannya. Sudah seharusnya ditetapkan sejak awal sebelum memulai perancangan basis data, atau paling tidak sebelum melaksanakan tahap implementasi basis data.

2.2.7 Basis Data

Basis Data adalah kumpulan informasi yang disimpan didalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer unutk memperoleh informasi dari basis data tersebut.

2.2.8 Sistem Basis Data

Basis data hanyalahsebuah objek yang pasif atau mati. Ia ada karena ada pembuatnya. Iatidak akan pernah berguna jika tidak ada pengelola atau penggeraknya.Yang menjadi penggeraknya secara langsung adalah program atauaplikasi (*software*). Gabungan antara program / aplikasi dengan basis data menghasilakn sebuah sistem. Karena itu, secara umum sebuah sistem basis data merupakan sistemyang terdiri atas kumpulan file (*table*) yang saling berhubungan (dalam sebuah basis data di sebuah sistem komputer) dan sekumpulan program

(DBMS) yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi file – file (table-tabel) tersebut.

Lebih jauh lagi, dalam sebuah sistem basis data, secara lengkapakan terdapat komponen – komponen utama sebagai berikut :

a. Perangkat Keras (*Hardware*)

Perangkat keras yang biasanya terdapat dalam sebuah sistem basis data adalah :

1. Komputer (satu untuk sistem yang stand – alone atau lebih dari satu untuk sistem jaringan).
2. Memori sekunder yang *online* (*harddisk*).
3. Memori sekunder yang *offline* (*Tape* atau *Removable Disk*)untuk keperluan *backup* data.
4. Media atau perangkat komunikasi (untuk sistem jaringan).

b. Sistem Operasi (*Operating System*)

Secara sederhana, sistem operasi merupakan program yang mengaktifkan atau memfungsikan sistem komputer, mengendalikan seluruh sumber daya (resource) dalam komputer dan melakukanoperasi – operasi dasar dalam computer (operasi I/O, pengelolaan file, dll). Program pengelola basis data hanya dapat aktif (running)jika sistem operasi yang dikehendakinya atau sesuai telah aktif.

c. Basis Data (*Database*)

Sebuah sistem basis data dapat memiliki beberapa basis data.Setiap basis data dapat berisi sejumlah objek basis data seperti file /table, indeks, dll. Di samping berisi atau menyimpan data, setiapbasis data juga mengandung definisi struktur (baik untuk basis datamaupun objek – objeknya secara detail).

d. Sistem (aplikasi / perangkat lunak) Pengelola Basis Data (DBMS).

Pengelolaan basis data secara fisik tidak dilakukan olehpemakai secara langsung, tetapi ditangani oleh sebuah perangkatlunak yang khusus. Perangkat lunak ini disebut DBMS (*Database Management*

Sistem) yang yang akan menentukan bagaimana datadiorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengamanan data, pemakaian data secarabersama, pemaksaan keakuratan atau konsistensi data, dan sebagainya.

e. Pemakai (*user*)

Ada beberapa jenis pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem :

1. Programmer Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui Data Manipulation Language (DML), yang disertakan(*embedded*) dalam program yang ditulis dalam bahasapemrograman induk (seperti C, Pascal, Cobol, dan lain lain).

2. User Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa menulismodul program. Mereka menyatakan query (untuk akses data dengan bahasa query yang disediakan oleh suatu DBMS.

3. User Umum (*End User*)

   Pemakai yang berinteraksi dengan sistem basis datamelalui pemanggilan satu program aplikasi permanen(executable program) yang telah ditulis atau disediakansebelumnya.

4. User Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non-konvensional, tetapi untuk keperluan – keperluan khusus,seperti untuk aplikasi AI, Sistem Pakar, Pengolahan Citra, dll,yang bisa saja mengakses basis data dengan / tanpa DBMSyang bersangkutan.

f. Aplikasi (perangkat lunak) lain (bersifat opsional)

Aplikasi (perangkat lunak) lain ini bersifat opsional. Artinya,ada atau tidaknya tergantung pada kebutuhan kita. DBMS yang kitagunakan lebih berperan salam pengorganisasian data dalam basisdata, sementara bagi pemakai basis data (khususnya yang

menjadi *end user / naive user*) dapat dibutuhkan / disediakan programkhusus / lain untuk melakukan pengisian, pengubahan danpengambilan data. Pemrogram ini ada yang sudah disediakanbersama dengan DBMS – nya, ada juga yang harus dibuat sendiridengan menggunakan aplikasi lain yang khusus untuk itu (*delevopment tools*).

2.2.9 SQL (Structured Query Language)

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de factomerupakan bahasa standar yang digunakan dalam manajemen basis data relasional.Saat ini hampir semua server basis data yang ada mendukung bahasa ini untukmelakukan manajemen datanya.

Secara umum, SQL terdiri dari dua bahasa, yaitu Data Definition Language(DDL) dan Data Manipulation Language (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umumimplementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI.

2.2.10 DBMS (Database Management System)

DBMS (DataBase Management System) adalah sistem yang secara khusus dibuat untuk memudahkan pemakai dalam mengelola basis data.Sistem ini dibuat untuk mengatasi kelemahan sistem pemrosesan yangberbasis berkas.Pada pendekatan yang berbasis berkas, umumnyaperancangan sistem didasarkan pada kebutuhan individual pemakai, bukanberdasarkan kebutuhan sejumlah pemakai. Setiap kali terdapat kebutuhanbaru dari seorang pemakai, kebutuhan segera diterjemahkan kedalamprogram komputer. Akibatnya, kemungkinan besar setiap program aplikasimenuliskan data tersendiri. Sementara itu ada kemungkinan data yangsama juga terdapat pada berkas-berkas lain yang digunakan oleh programaplikasi lain.

DBMS merupakan perantara bagi pemakai dengan basis data.Untuk merinteraksi dengan DBMS (basis data) menggunakan bahasa basisdata yang telah ditentukan oleh perusahaan DBMS. Bahasa basis databiasanya terdiri atas perintah-perintah yang di formulasikan sehinggaperintah tersebut akan diproses olah DBMS. Perintah-perintah biasanyaditentukan oleh user. Ada 2 bahasa basis data:

2.2.11 DDL (Data Definition Language)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data danobjek-objek yang diperlukan dalam basis data, misalnya tabel, view, user, dansebagainya. Secara umum, DDL yang digunakan adalah CREATE untuk membuat objek baru, USE untuk menggunakan objek, ALTER untuk mengubah objek yangsudah ada, dan DROP untuk menghapus objek. DDL biasanya digunakan olehadministrator basis data dalam pembuatan sebuah aplikasi basis data.

CREATE

CREATE digunakan untuk membuat basis data maupun objek-objek basis data. SQL

yang umum digunakan adalah:

“CREATE DATABASE nama_database”

“CREATE TABLE nama_tabel”

2.2.12 DML (Data Manipulation Language)

` DML digunakan untuk melakukan menipulasi dan pengambilandata pada suatu basis data seperti penambahan data baru ke dalam basisdata, menghapus data dari suatu basis data dan pengubahan data di suatubasis data. Dalam pembuatan DBMS diperlukan beberapa komponenfungsional penyusunnya sebagai berikut:

a. DML *Precompiler* yaitu mengkonversi pernyataan-pernyataanDML yang dimasukkan di dalam program aplikasi ke dalampemanggilan prosedur normal di dalam bahasa induknya.Procompiler harus

berinteraksi dengan query processor untukmembuat kode-kode yang diperlukan.

b. *Query Processor* yaitu menterjemahkan pernyataan-pernyataanbahasa query ke dalam instruksi-instruksi *low-level* yangdimengerti oleh database *manager*.
c. DDL *Compiler* yaitu mengkonversi pernyataan DDL ke dalamsekumpulan tabel yang mengandung metadata atau "datamengenai data".
d. *Database Manager* yaitu menyediakan *interface* antara data*low-level* yang disimpan di dalam basis data dengan program-program aplikasi dan *queries* yang dikirimkan ke sistem.

Salah satu tujuan dari DBMS adalah memberikan tampilan kepada pengguna dalam hal menyampaikan data. Untuk itu dalam DBMS terdapatLevel Abstraksi Data. Level ini berguna untuk menyembunyikan detailatau kompleksitasnya basis data seperti bagaimana data disempan dandiolah. Sehingga pengguna hanya melihat tampilan yang dibutuhkan olehpengguna. Contoh dari DML adalah *select* * pelanggan *where name* = 'ade'.

2.2.13 Level Fisik

Level fisik merupakan level yang paling bawah. Pada level ini memperlihatkan bagaimana sesungguhnya data disimpan.

1. **Level Konseptual**

   Level ini menggambarkan bagaimana sebenarnya basis datadisimpan dan berhubungan dengan data lainnya.

2. **Level View**

   Level abstaraksi ini hanya menunjukkan sebagian dari basis data.Pada umumnya pengguna tidak melibatkan secara langsung sehinggapengguna hanya melihat data sesuai dengan yang dibutuhkan.

2.2.14 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik / gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*).

UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software. UML adalah salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented.

Sedangkan menurut para ahli, pengertian UML adalah :

1. Menurut (Hend, 2006) "Unified Modeling Language (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak".
2. Menurut (Adi Nugroho : 2005). "Unified Modeling Language (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek".

2.2.15 Benda / *Things*

Adalah hal yang sangat mendasar dalam model UML, juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari sebuah konsep dan atau fisik.

Bentuk dari beberapa benda/thing adalah sebagai berikut :

1. *Classes*, yang diuraikan sebagai sekelompok dari object yang mempunyai atribute, operasi, hubungan yang semantik. Sebuah kelas mengimplementasikan 1 atau lebih interfaces. Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, yang mempunyai sebuah nama, atribute, dan metode pengoperasiannya.
2. *Interfaces*, merupakan sebuah antar-muka yang menghubungkan dan melayani antar kelas dan atau elemen. '*Interface*' / antar-muka

mendefinisikan sebuah set/kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah antar-muka berdiri sendiri dan umumnya merupakan pelengkap dari kelas atau komponen.

3. *Collaboration*, yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas / elemen-elemen yang bekerja secara bersama-sama. *Collaborations* mempunyai struktur dan dimensi. Pemberian sebuah kelas memungkinkan berpartisipasi didalam beberapa collaborations dan digambarkan dengan sebuah 'elips' dengan garis terpotong-potong.
4. *Use cases*, adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. '*use case*' digunakan untuk membentuk tingkah-laku benda / things dalam sebuah model serta direalisasikan oleh sebuah *collaboration*. Umumnya '*use case*' digambarkan dengan sebuah '*elips*' dengan garis yang solid, biasanya mengandung nama.
5. *Nodes*, merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalaha sebuah komputer, umumnya mempunyai sedikitnya memory dan prosesor. Sekelompok komponen mungkin terletak pada sebuah *node* dan juga mungkin akan berpindah dari *node* satu ke *node* lainnya. Umumnya *node* ini digambarkan seperti kubus serta hanya mengandung namanya

2.2.16 Hubungan / Relationship

Ada empat macam hubungan di dalam penggunaan UML, diantaranya adalah sebagai berikut :

1. *Dependency*, adalah hubungan semantik antara dua benda/*things* yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula. Umumnya sebuah *dependency* digambarkan sebuah panah dengan garis terputus-putus.

2. *Association*, hubungan antar benda struktural yang terhubung diantara objek. Kesatuan objek yang terhubung merupakan hubungan khusus, yang menggambarkan sebuah hubungan struktural diantara seluruh atau sebagian. Umumnya *assosiation* digambarkan dengan sebuah garis yang dilengkapi dengan sebuah label, nama, dan status hubungannya.
3. *Generalizations* adalah menggambarkan hubungan khusus dalam obyek anak/*child* yang menggantikan obyek *parent* / induk. Dalam hal ini, obyek anak memberikan pengaruhnya dalam hal struktur dan tingkah lakunya kepada objek induk. Digambarkan dengan garis panah.
4. *Realizations*, merupakan hubungan semantik antara pengelompokkan yang menjamin adanya ikatan diantaranya. Hubungan ini dapat diwujudkan diantara interface dan kelas atau elements, serta antara *use cases* dan *collaborations*. Model dari sebuah hubungan *realization*.

2.2.17 Bagan / Diagram

UML sendiri terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu; *use-case, class, object, state, sequence, collaboration, activity, component,* dan *deployment* diagram.

1. *Use Case Diagram*, menggambarkan sekelompok use cases dan aktor yang disertai dengan hubungan diantaranya. Diagram use cases ini menjelaskan dan menerangkan kebutuhan / requirement yang diinginkan / dikehendaki user/pengguna, serta sangat berguna dalam menentukan struktur organisasi dan model dari pada sebuah sistem.
2. *Class Diagram* adalah bagan yang memperlihatkan struktur statis dari kelas aktual di dalam sistem.
3. *Object Diagram*, yang merupakan varian dari kelas diagram yang memperlihatkan lebih detail banyaknya obyek yang mengintantiasi (*instances*) kelas.

4. *State Diagram*, yang memperliatkan semua keadaan (*state*) yang dapat dimiliki oleh kelas dan *event* yang dapat merubah keadaan tersebut.
5. *Sequence Diagram*, yang memperlihatkan kolaborasi dinamis antara objek-objek dengan suatu urutan pesan (*a sequence of message*) antar objek tersebut.
6. *Collaboration Diagram* adalah bagan yang memperlihatkan kolaborasi dinamis antar objek tanpa memperhatikan aspek waktu.
7. *Activity Diagram* adalah bagan yang memperlihatkan aliran urutan aktifitas.
8. *Component Diagram* adalah bagan yang memperlihatkan struktur fisik dari *source code* dalam *terminology code components.* Komponen berisi informasi tentang *logical class* dapat berupa komponen *source code*, komponen biner atau komponen yang dapat dieksekusi.
9. *Deployment Diagram* adalah bagan yang memperlihatkan arsitektur fisik dari *hardware* dan *software* pada sistem.

2.2.18 Desain Perangkat Lunak

Menurut *Whitten et al*(2004), Desain perangkat lunak yaitu tahapan atau aktivitas yang difokuskan pada spesifikasi detil dari solusi berbasis komputer. Desain perangkat lunak sering juga disebut sebagai *physical design*. Jika tahapan analisis sistem menekankan pada masalah bisnis (*businessrule*), maka sebaliknya disain perangkat lunak fokus pada sisi teknis danimplementasi sebuah perangkat lunak.

Output utama dari tahapan desain perangkat lunak adalah spesifikasi disain. Spesifikasi ini meliputi spesifikasi desain umum yangakan disampaikan kepada *stakeholder* sistem dan spesifikasi desain rinciyang akan digunakan pada tahap implementasi. Spesifikasi desain umumhanya berisi gambaran umum agar *stakeholder* sistem mengerti akanseperti apa perangkat lunak yang akan dibangun. Biasanya diagram USDtentang perangkat lunak yang baru merupakan point penting dibagian ini.Spesifikasi disain rinci atau kadang disebut desain arsitektur rinciperangkat lunak diperlukan untuk merancang sistem sehingga

memilikikonstruksi yang baik, proses pengolahan data yang tepat dan akurat,bernilai, memiliki aspek user friendly dan memiliki dasar-dasar untukpengembangan selanjutnya.

Desain arsitektur ini terdiri dari desain database, desain proses,desain *user interface* yang mencakup desain *input*, *output form* danreport, desain hardware, software dan jaringan. Desain proses merupakankelanjutan dari pemodelan proses yang dilakukan pada tahapan analisis.

2.2.19 Bahasa Pemrograman PHP

PHP (Personal Home Page) adalah server-side programming yang popular digunakan untuk membuat web-based application. PHP dikenal sebagai sebuah bahasa scripting yangmenyatu dengan tag-tag HTML yang diekseskusi di server dan digunakan untuk membuathalaman webyang dinamis seperti guestbook, statistik pengunjung, polling, email, dan masihbanyak lagi.

Adapun keunggulan yang dimiliki oleh bahasa pemrograman PHP adalah sebagaiberikut :

a. *Life Cycle* yang sangat singkat , sehingga PHP selalu *up to date* mengikutiperkembangan teknologi internet.
b. *Cross Platform*, yakni PHP dapat dipakai di hampir semua webserver yang ada dipasaran (terutama Apache dan Microsoft IIS) dan dijalankan pada berbagai sistemoperasi (Linux, Windows, FreeBSD).
c. PHP mendukung koneksi ke banyak *database* baik yang gratis maupun komersil,seperti MySQL, mSQL, Oracle, Microsoft SQL Server, Interbase, dan banyak lagi.
d. PHP bersifat open source dan gratis.
e. Kemudahan dalam mendapatkan dokumentasi. di Internet, kita tidak akan sulit untukmencari baik itu referensi, kode-kode PHP yang sudah jadi dan juga mengajukanpertanyaan pada grup-grup diskusi yang di dalamnya banyak sekali para master PHP.

2.2.20 Pengenalan MySQL

MySQL merupakan sebuah software yang berguna sebagai suatu Database Server yang cukup terkenal. Kepopulerannya seiring dengan penggunaan script PHP untuk *web programming*. *Database* server itu sendiri merupakan suatu software yang bertugas untuk melayani permintaan (*request*) *query* dari *client*.

MySQL sebagai suatu *database* server mempunyai beberapa kemampuan, salah satunya harus menyediakan suatu sistem manajemen *database* yang dapat mengatur bagaimana menyimpan, menambah, mengakses data dan transaksi-transaksi database lainnya.MySQL cepat sekali berkembang, karena MySQL merupakan suatu *software* yang *Open Source.*

2.2.21 Pengenalan phpDesigner 8.0

PHP Designer 2012 atau PHP Designer Versi 8 aplikasi untuk mengedit PHP/HTML. Mungkin selama ini bagi para *coder* khususnya untuk *web developer*, program *editor* Adobe Dreamweaver pasti sangat membantu, dan untuk pemula yang baru belajar rasanya PHP Designer 8 adalah Code editor yang lebih membantu karena program ini lebih intuitif dengan memberitahu kesalahan kode, prosedur menggunakan *syntax* tertentu, dan satu lagi yang masih jarang di temui yaitu 'PHP run' dimana kita dapat lansung mem-*preview* kode PHP kita, dan juga IDE (*integrated development environment*) ini juga mendukung OOP(*object oriented programming*) pada PHP dan memudahkan pemrograman web 2.0.

PHP designer mendukung berbagai macam bahasa pemograman web mulai dari PHP, HTML, CSS, XML, Java Script, SQL, VBS, Perl, bahkan C#, Python, dan Ruby juga turut di dukung.

Keuntungan bagi programer PHP pemula, PHP designer akan memunculkan jendela *pop-up* saat kita mengetikan kode yang menampilkan petunjuk mengunakan *syntax* yang kita ketikan, seperti parameter apa yang bisa di masukan dan sebagainya. selain itu program ini secara *live* akan menunjukan di baris mana kita melakukan kesalahan penulisan *syntax* dan fitur *error checking* PHP designer lebih baik jika di bandingkan dengan Dreamweaver.

2.2.22 Pengenalan XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program.

Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

Dengan menginstal XAMPP, kita tidak perlu menginstal aplikasi server satu persatu karena di dalam XAMPP sudah terdapat :

a. Apache 2.2.17
b. MySQL 5.5.8 (Community Server)
c. PHP 5.3.5 (VC6 X86 32bit) + PEAR
d. XAMPP Control Version 2.5 from www.nat32.com
e. XAMPP Security
f. SQLite 2.8.15
g. OpenSSL 0.9.8o
h. PhpMyAdmin 3.3.9
i. ADOdb 5.11
j. Mercury Mail Transport System v4.62
k. FileZilla FTP Server 0.9.37
l. Webalizer 2.01-10
m. Zend Optimizer 3.3.0
n. Perl 5.10.1
o. Mod_perl 2.0