

BAB 2

TINJAUAN PUSTAKA

2.1. Profil Perusahaan

Profil PT. GALMA meliputi berbagai aspek yaitu sejarah, visi dan misi, struktur organisasi dan deskripsi kerja

2.1.1. Sejarah Perusahaan

Pada awalnya CV. GALMA lebih banyak berkecimpung dalam bidang teknik Sipil. Dalam perkembangannya, terutama didorong oleh tuntutan profesionalisme peningkatan pelayanan kerja, CV. GALMA mengalami beberapa kali perubahan manajemen dan yang terakhir pada 2011. Tujuan utama perombakan manajemen tersebut selain sebagai upaya restrukturisasi ke dalam tubuh perusahaan guna memperlancar kenaikan jenjang dan peningkatan profesi tenaga ahli dan tenaga pendukungnya, juga merupakan salah satu upaya peningkatan kualitas profesionalisme pelayanan para mitra dan rekanan.

Dalam pelayanan konsultan teknik CV. GALMA telah bekerjasama dalam bentuk Joint Operational (Konsorsium) terutama dengan sesama konsultan nasional sebagaimana anjuran pemerintah dalam rangka peningkatan daya saing usaha dalam negeri serta mutu pekerjaan, juga menghadapi ERA GLOBALISASI. Sampai saat ini CV. GALMA telah menangani berbagai macam proyek baik dari instansi pemerintah maupun yang dibiayai oleh badan International.

2.1.2. Visi dan Misi Perusahaan

Visi

Menjadi Perusahaan Nasional yang Berkualitas dan Inspiratif.

Misi

1. Memberi pelayanan jasa konsultasi, penelitian dan pengembangan terbaik, bermutu tinggi, berkualitas dan tepat waktu.

1. Membantu klien mengembangkan ide-ide kreatif, gagasan inovatif, pemikiran inspiratif dan memberikan solusi secara brilian.
2. Meningkatkan kualitas sumber daya manusia untuk menjadi konsultan dan peneliti yang mampu memberikan pelayanan jasa terbaik.

2.1.1. Logo Perusahaan

Logo PT GALMA dilihat pada Gambar 2.1

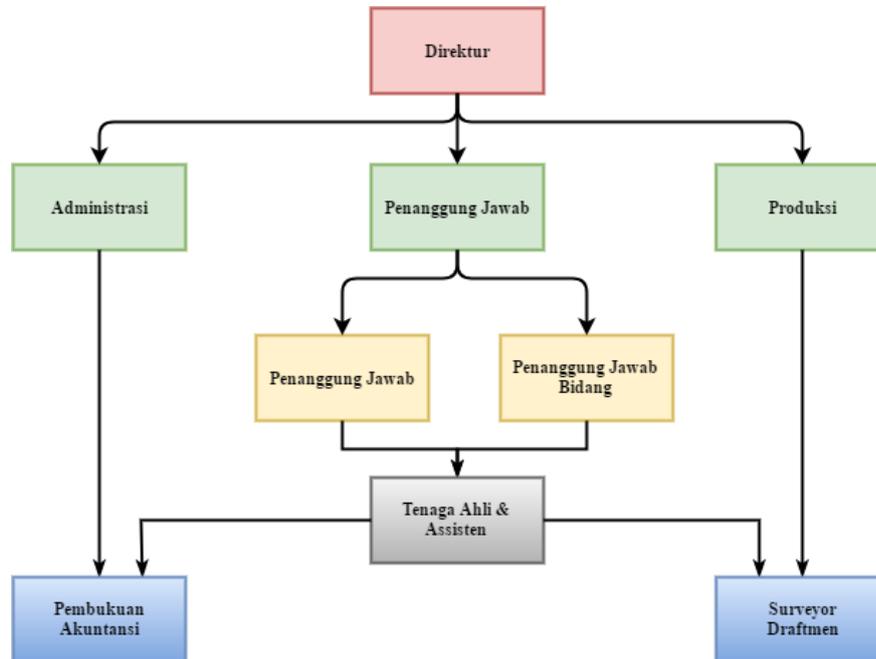


Gambar 2.1 PT Galma

Lambang PT GALMA memiliki arti sebagai berikut:

1. Representasi dari konsep perusahaan yang dinamis.
2. Profesionalitas serta menginspirasi kepercayaan kepada seluruh pemangku kepentingan.

2.1.1. Struktur Organisasi



Gambar 2.2 Struktur Organisasi PT GALMA

Deskripsi Kerja:

1. Direktur

Seseorang yang memiliki perusahaan tersebut atau orang profesional yang ditunjuk menjalankan dan memimpin perusahaan.

- a. Memimpin perusahaan dengan menertibkan kebijakan perusahaan atau intitusi
- b. Menyampaikan laporan kepada kinerja perusahaan
- c. Menyetujui anggaran tahunan perusahaan

2. Administrasi

Usaha atau kegiatan yang berkenaan dengan penyelenggaraan kebijakan untuk mencapai tujuan.

- a. Pembuatan laporan keuangan PT GALMA
- b. Mengisi data kepegawaian, pelaksana, menyimpan data kepegawaian karyawan
- c. Membantu direktur terutama dalam hal keuangan dan sumber daya

1. Penanggung Jawab
 - a. Memberikan saran,kritik atau ide kepada Direktur
 - b. Memberikan bantuan secara moril mapupun materil
 - c. Mencari solusi serta menyelesaikan bila terjadi permasalahan di PT GALMA
 - d. Bertanggung jawab atas kegiatan di perusahaan
2. Produksi
 - a. Briefing mengenai menyiapkan material
 - b. Menjaga kualitas produk
 - c. Memonitor kondisi produksi
 - d. Membuat laporan produksi
3. Penanggung Jawab Bidang
 - a. Mengkoordinasi organisasi dalam pelaksanaan kegiatan seluruh bidang kegiatan organisasi
 - b. Mewakili organisasi untuk menghadiri rapat atau agendan
 - c. Bertanggung jawab atas permasalahan di perusahaan
4. Pembukuan Akuntansi
 - a. Melakukan aktifitas pencataan data usaha suatu perusahaan
 - b. Bertanggung jawab terhadap seluruh catatan usaha atau operusahaan
5. Tahap ahli & Assisten
 - a. Mengkoordinasi operasi jadwal dan sasaran dipenuhi
 - b. Berkomunikasi dengan client dan mengevaluasi kebutuhan
 - c. Membuat laporan analisis data
 - d. Mengawasi staff
 - e. Membantu direktur dalam merencanakan dan menerepatkan strategi
6. Surveyor Draftmen
 - a. Membuat gambar pelaksaan
 - b. Menyesuaikan gambar perencanaan dengan kondisi nyata lapangan
 - c. Menjelaskan kepada tenaga ahli lapangan

2.2. Landasan Teori

2.2.1. Data

Data adalah kumpulan fakta yang diperoleh dari suatu hasil pengukuran. Suatu pengambilan keputusan yang baik merupakan hasil dari penarikan kesimpulan yg didasarkan pada data atau fakta yang akurat. Untuk mendapatkan data yang akurat diperlukan lah alat ukur yang disebut Instrumen. Alat ukur atau instrumen adalah alat ukur/Instrumen yang VALID dan RELIABE[2].

2.2.2. Basis Data

Basis data (Database) adalah sekumpulan informasi yang diorganisasikan ke dalam tata cara dan aturan yang khusus.

Database adalah kumpulan informasi yang ditata dan merupakan kesatuan yang utuh yang disimpan di dalam perangkat keras secara sistematis sehingga dapat diolah kembali menggunakan perangkat lunak. Dengan sistem yang digunakan disini data yang terhimpun dalam database dapat menghasilkan informasi yang berguna bagi penggunanya.

Operasi dasar *Basis Data*

Ada point operasi basis data dibawah ini diantaranya :

- a. Pembuatan basis data baru (create database, yang identik dengan pembuatan lemari arsip baru
- b. Penghapusan basis data (drop database), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya jika ada)
- c. Pembuatan file/tabel baru ke suatu basis data (create table), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada
- d. Penghapusan file/tabel dari suatu basis data (drop table), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip
- e. Penambahan/pengisian data baru ke sebuah file/tabel du sebuah basis data (insert), yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
- f. Pengambilan data dari sebuah file/tabel (retrieve/searce), yang identik

dengan pencarian lembaran arsip dari sebuah map arsip

- g. Perubahan data dari sebuah file/tabel (update), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip
- h. Penghapusan data dari sebuah file/tabel (delete) yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip[3].

2.2.3. Data Mining

Data Mining adalah suatu proses penambangan informasi yang penting dari satu data. Informasi penting ini dapat dari suatu proses yang amat rumit seperti menggunakan artificial intelligence, Istilah *data mining* memiliki hakikat sebagai disiplin ilmu yang tujuan utamanya adalah untuk menemukan, menggali, atau menambang pengetahuann dari data atau informasi yang kita miliki. Data mining sering disebut juga disebut sebagai Knowledge Discovery in Database (KDD), KDD disini adalah kegiatan yang meliputi pengumpulan, pemakaian data untuk menemukan aturan, pola atau hubungan dalam set data berukuran besar. Berikut adalah proses *data mining* antara lain[4]:

1. Deskripsi

Deskripsi disini bertujuan untuk mencoba apakah cara untuk mendeskripsikan pola yang tersembunyi dalam data.

2. Estimasi

Estimasi bertujuan untuk lebih kearah numerik dari segi kategori

3. Prediksi

Prediksi bertujuan untuk prediksi hasil yang menunjukkan sesuatu yang belum terjadi (mungkin terjadi di masa depan)

4. Klarifikasi

Klarifikasi bertujuan yang bersifat kategorik. Misalnya, kita akan mengklarifikasi pendapatan dalam tiga kelas, yaitu pendapatan tinggi, pendapatan rendah, pendapatan sedang

5. Clustering

Clustering lebih kea rah pengelompokan *record*, pengamatan, atau kasus dalam keals yang memiliki kemiripan.

1. Asosiasi

Asosiasi disini bertujuan mengidentifikasi hubungan antara berbagai kejadian yang terjadi pada satu waktu.

2.2.4. Metode Data Mining (CRIPS-DM)

Metode CRIPS-DM merupakan siklus hidup proyek pada data mining yang terbagi dalam enam fase seperti terlihat pada gambar 1.1 di bawah ini. Pengerjaan menyesuaikan dengan kasus yang diambil, dimana setiap tahapan bergantung pada keluaran fase sebelumnya. Fase yang sudah dikerjakan mungkin dapat kembali pada fase sebelumnya apabila diperlukan evaluasi atau kesalahan pada fase sebelumnya[1].

Berikut penjelasan dari setiap fase CRISP-DM:



Gambar 2.3 Proses Data Mining menurut CRIPS-DM

1. Fase Pemahaman Bisnis (Business Undersanting)
 - a) Penentuan tujuan proyek dan kebutuhan secara detail dalam lingkup bisnis atau unit penelitian secara keseluruhan.
 - b) Menerjemahkan tujuan dan batasan menjadi formula dari permasalahan data mining.
 - c) Menyiapkan strategi awal untuk mencapai tujuan.

1. Fase Pemahaman Data (Data Understanding Phase)
 - a) Mengumpulkan data.
 - b) Menganalisis data.
 - c) Mengevaluasi kualitas data.
 - d) Memilih sebagian data yang mengandung pola dari permasalahan.
2. Fase Pengolahan Data (Data Preparation Phase)
 - a) Menyiapkan data awal, kumpulan data yang akan digunakan untuk keseluruhan fase berikutnya.
 - b) Memilih kasus dan variable yang ingin dianalisis dan yang sesuai analisis yang akan dilakukan.
 - c) Lakukan perubahan pada beberapa variable jika dibutuhkan.
 - d) Menyiapkan data awal agar siap untuk pemodelan.
3. Fase Pemodelan (Modeling Phase)
 - a) Pilih dan aplikasi teknik pemodelan yang sesuai.
 - b) Mengatur model untuk mengoptimalkan hasil.
 - c) Proses dapat kembali ke fase pengolahan data untuk menjadikan data ke dalam bentuk yang sesuai dengan spesifikasi kebutuhan teknik data mining tertentu.
4. Fase Evaluasi (Evaluation Phase)
 - a) Mengevaluasi satu atau lebih digunakan dalam fase pemodelan untuk mendapatkan kualitas dan efektivitas.
 - b) Menetapkan apakah terdapat model yang memenuhi tujuan pada fase awal.
 - c) Menentukan apakah terdapat permasalahan penting dari bisnis atau penelitian yang tidak tertangani dengan baik.
 - d) Mengambil keputusan berkaitan dengan penggunaan hasil dari data mining.

1. Fase Penyebaran (Deployment Phase)

- a) Menggunakan model yang dihasilkan. Terbentuknya model tidak menandakan telah terselesaikannya proyek.
- b) Contoh sederhana penyebaran: pembuatan laporan keuangan.

2.2.5. Tahapan Asosiasi

Tahapan Asosiasi bertujuan untuk menganalisis pola asosiasi akan ditemukan atribut-atribut yang menunjukkan kondisi dimana atribut-atribut tersebut sering muncul bersamaan dalam suatu data yang diberikan. Penggalian pola asosiasi ini biasanya sering digunakan dalam menganalisis data transaksi. Ada beberapa algoritma yang digunakan dalam upaya untuk memperbaiki kinerja panggilan pola asosiasi tapi, algoritma-algoritma tersebut hanya mampu menangani kondisi tertentu saja dan memiliki kekurangan dalam kondisi lainnya. Sedangkan, dalam penggalian data efektifitas dan efisiensi sangatlah diperlukan[5].

2.2.6. Assosiation Rule

Assosiation Rule Mining adalah sebuah teknik adata *mining* untuk menemukan aturan asosiatif antara suatu kombinasi sebuah item.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan cara dua paramater, yaitu *support* dan *confidence*. *Support* (nilai penunjang) adalah sebuah presentase kombinasi dari *item* tersebut dalam *database*, sedangkan *confidence* (nilai kepastian adalah kuatnya hubungan antar-item dalam aturan assosiasi).

Metode dasar dari analisis asosiaasi terbagi menjadi dua tahap :

1. Analisis pola frekuensi tinggi

Tahapan ini mencari sebuah kombinasi item yang sudah memenuhi syarat minimum dari nilai *support* dalam database. Nilai *support* sebuah item diperoleh dengan rumus, *support* dari suatu association rule adalah presentasi kombinasi itemtersebut dalam database,

dimana jika mempunyai item A dan item B maka support adalah proporsi dari data transaksi dalam database yang mengandung A dan B. Rumus untuk menghitung nilai *support* dari dua item adalah sebagai berikut [6]:

$$\text{Support}(A) = \frac{\text{Jumlah transaksi mengandung } A}{\text{Total Transaksi}} \quad (1)$$

Sementara itu, nilai *support* dari 2 item

diperoleh dari rumus berikut.

$$\text{Support}(A, B) = P(A \cap B) \quad (2)$$

$$\text{Support}(A, B) = \frac{\# \text{Transaksi mengandung } A \text{ dan } B}{\# \text{Transaksi}} \quad (3)$$

1. Pembentukan Aturan Asosiasi

Setelah pola frekuensi tinggi ditemukan, selanjutnya dicari aturan dari asosiasi yang cukup kuat tingkat keterkaitannya antar *item* dalam *antecedent* (pendahulu) dan *consequent* (pengikut) serta memenuhi syarat minimum untuk *confidence* dengan menghitung *confidence* aturan Asosiatif .

Misalkan D adalah himpunan transaksi, dimana setiap transaksi T dalam D merepresentasikan himpunan *item* yang berada dalam I. I adalah himpunan *item* yang dijual. Misalkan kita memilih himpunan *item* A dan himpunan *item* lain B, kemudian aturan asosiasi akan berbentuk:

$$\text{Jika } A, \text{ maka } B(A \rightarrow B)$$

Dimana *antecedent* A dan *consequent* B yang merupakan *subset* dari I, dan A dan B merupakan *mutually exclusive* dimana aturan :

$$\text{Jika } A, \text{ maka } B$$

Tidak berarti

Jika B, maka A

Sebuah *itemset* adalah himpunan *item-item* yang ada dalam I, dan *k-itemset* adalah *itemset* yang berisi *k item*. *Frekuensi itemset* merupakan *itemset* yang memiliki frekuensi kemunculan lebih dari nilai minimum yang telah ditentukan (ϕ). Misalkan $\phi = 2$, maka semua *itemset* yang frekuensi kemunculannya lebih dari atau sama dengan 2 kali disebut *frequent*. Himpunan dari *frequent k-itemset* dilambangkan dengan F_k .

Nilai *confidence* dari aturan $A \rightarrow B$ diperoleh dari rumus berikut.

$$Confidence = P(B | A) = \frac{\Sigma \text{Transaksi mengandung A dan B}}{\Sigma \text{Transaksi mengandung A}} \quad (4)$$

2.2.7. Frequent Patters

Frequent pattern growth (FP-growth) adalah algoritma untuk menemukan sejumlah frequent itemset dari data transaksi yang tersimpan dan tersusun dalam basis data. Dalam penelitian ini algoritma Apriori dan frequent pattern growth (FP-growth) yang digunakan untuk membantu menemukan sejumlah aturan asosiasi dari suatu basis data[7].

2.2.8. Algoritma Frequent Pattern Growth (FP-Growth)

Algoritma Frequent Pattern-Growth atau yang biasa disebut dengan FPGrowth merupakan pengembangan dari algoritma Apriori, sehingga dalam algoritma FP-Growth ini, segala kekurangan dalam aloritma Apriori telah diperbaiki .

Frequent Pattern Growth atau (FP-Growth) adalah suatu salah satu alternatif algoritma yang digunakan untuk menentukan himpunan data mana yang paling sering muncul (frequent itemset) dalam sebuah kumpulan dataset .

Karakteristik dari algoritma FP-Growth adalah struktur data nya yang digunakan adalah tree atau pohon yang disebut dengan FPTree. Dengan

menggunakan FP-Tree, algoritma FP-Growth dapat langsung mengekstrak frequent itemset dari FP-Tree[8].

1.2.1.1. Langkah-Langkah Proses Perhitungan Association Rule Dengan Algoritma FP-Growth

Proses suatu perhitungan association rule terdiri dari beberapa tahap seperti di bawah ini sebagai berikut:

1. Membuat Header Item

Header dalam hal ini selain sebagai header suatu item ke FP-Tree juga sebagai jenis item dasar yang memenuhi minimum support. Setelah mendapatkan item dan nilai support-nya, maka item yang tidak *Frequent* dibuang dan item diurutkan berdasarkan nilai support-nya. Header untuk item, disiapkan pada suatu array tertentu dan ditambahkan ketika membuat pembentukan FP-Tree.

2. Membuat FP-Tree

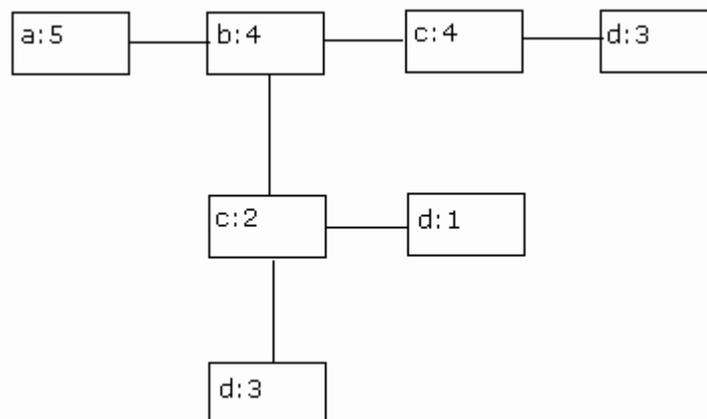
FP-Tree dibangun dengan mencari item sesuai urutan pada item yang *Frequent* Data transaksi tidak perlu diurutkan, dan untuk tiap item yang ditemukan bisa langsung dimasukkan ke dalam FP-Tree. Sesudah membuat root, tiap item yang ditemukan dimasukkan berdasarkan path pada FP-Tree. Jika item yang ditemukan sudah ada, maka nilai support item tersebut yang ditambahkan. Namun jika path belum ada, maka dibuat node baru untuk melengkapi path baru pada FP-Tree tersebut. Hal ini dilakukan selama item pada transaksi masih ada yang qualified, artinya memenuhi nilai minimum support. Jadi, item-item yang ditemukan dalam transaksi akan berurutan memanjang ke bawah. Dalam struktur FP-Tree, diterapkan alur path dari child hingga ke root. Jadi, suatu path utuh dalam FP-Tree adalah dari child terbawah hingga ke root. Tiap node pada FP-Tree memiliki pointer ke parent, sehingga pencarian harus dimulai dari bawah.

3. Pattern Extraction

Pattern extraction dilakukan berdasarkan keterlibatan item pada suatu path. Di setiap path, diperiksa semua kombinasi yang mungkin dimana item tersebut terlibat. Di iterasi berikutnya dilakukan dengan melibatkan item berikutnya, tanpa melibatkan item sebelumnya, sehingga pattern 28 yang sama tidak akan ditemukan dua kali pada path yang sama. Bila item pertama suatu hasil kombinasi bukan item terakhir (sebelum root), maka kombinasi itemset tersebut masih bisa dikembangkan lagi.

1. Memasukkan setiap pattern yang ditemukan ke dalam Pattern Tree

Setelah mengolah *FP-Tree* menjadi pattern-pattern, diperlukan proses akumulasi pattern-pattern yang ditemukan mengingat pattern yang sama dapat ditemukan pada path yang berbeda. Untuk itu digunakan struktur data *Pattern Tree* (lihat Gambar 2.3). Setiap node di *Pattern Tree* merepresentasikan dan menyimpan frekuensi suatu pattern. *Pattern Tree* terdiri atas *Pattern Tree Node* yang menyimpan nilai item, nilai support dan dilengkapi dengan dua pointer yaitu untuk horisontal dan vertikal.



Gambar 2.4 Pattern Tree

Misalnya pada node d:1 di atas, berarti terdapat pattern a-c-d bernilai support 1. Kemudian bila ada pattern a-c-d lagi bernilai support n yang ditemukan dari *FP-Tree* maka nilai support 1 tersebut menjadi n+1. Contoh hasil lengkap dari *PatternTree* tersebut:

1. a:5 menggambarkan bahwa ada pattern a sebanyak 5
2. b:4 menggambarkan bahwa ada pattern a-b sebanyak 4
3. c:4 menggambarkan bahwa ada pattern a-b-c sebanyak 4
4. d:3 menggambarkan bahwa ada pattern a-b-c-d sebanyak 3
5. c:2 menggambarkan bahwa ada pattern a-c sebanyak 2
6. d:1 menggambarkan bahwa ada pattern a-c-d sebanyak 1
7. d:3 menggambarkan bahwa ada pattern a-d sebanyak 3

1. Mengurutkan dan Menyeleksi Pattern

Pattern yang tidak memenuhi minimum support, dihapus dari daftar pattern. Pattern-pattern yang tersisa kemudian diurutkan untuk memudahkan pembuatan rules.

2.2.9. Object Oriented Programming (OOP)

OOP (Object Oriented Programming) adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi atau berbasis pada sebuah obyek dalam pembangunan sebuah program aplikasi, maksudnya disini bahwa orientasi pembuatan program tidak perlu lagi menggunakan orientasi yang linear melainkan berorientasi pada objek-objek yang terpisah-pisah. Suatu perintah dalam bahasa ini diwakili oleh sebuah Obyek yang didalamnya berisi beberapa perintah-perintah standar sederhana[9].

2.2.10. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti

lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C[10].

2.2.6.1. *Diagram Use Case*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu[10].

2.2.6.2. *Diagram Kelas*

Class adalah sebuah spesifikasi yang digunakan apabila diinstansiasi akan menghasilkan sebuah objek, yang merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi)[10].

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

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metode

2.2.6.3. *Diagram Aktivitas*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal,

decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum[10].

2.2.6.1. Diagram Objek

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalan objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus objek, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggung jawabkan. Diagram objek juga berfungsi untuk mendefinisikan contoh nilai atau isi dari alat ribut tiap kelas[10].

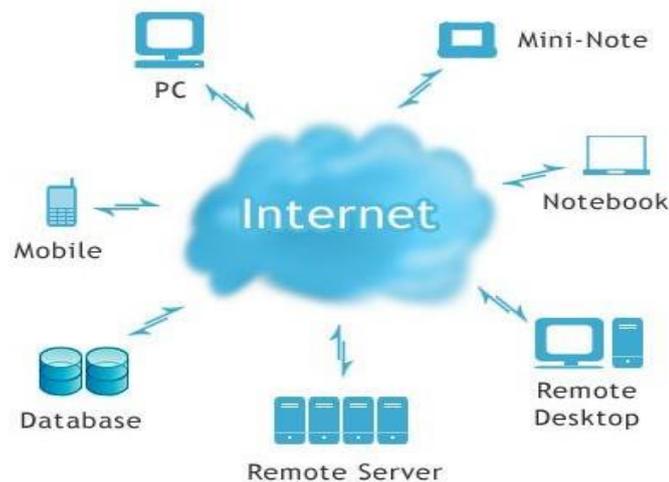
2.2.11. PHP

PHP adalah bahasa pemrograman yang berfungsi untuk membuat website yang dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga dalam membuat PHP bisa menampilkan konten yang lebih dinamis dari sebuah website. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”[11].

2.2.12. Cloud Computing

Dalam kata lain *cloud computing* bisa berarti akses fasilitas komputer secara bersama-sama memalui Internet dari berbagai lokasi. Sebagai contoh, sebuah Bank besar menggunakan *cloud computing* untuk operasional transaksi

online, dan secara tidak disadari sebagian dari kita sebenarnya telah menggunakan fasilitas *cloud computing* dalam bentuk *email* dan *World Wide Web (WWW)*[12].



Gambar 2.5 Cloud Computing

Karakteristik *Cloud Computing*

Lima karakteristik penting dari *cloud computing*, yaitu :

1. *On-demand self-service*. Konsumen dapat menentukan kemampuan komputasi secara sepihak, seperti *server time* dan *network storage*, secara otomatis sesuai kebutuhan tanpa memerlukan interaksi manusia dengan masing-masing penyedia layanan.
2. *Broad network access*. Kemampuan yang tersedia melalui jaringan dan diakses melalui mekanisme standar yang mengenalkan penggunaan berbagai platform (misalnya, telepon selular, tablets, laptops, dan workstations).
3. *Resource pooling*. Penyatuan sumberdaya komputasi yang dimiliki penyedia untuk melayani beberapa konsumen virtual yang berbeda, ditetapkan secara dinamis dan ditugaskan sesuai dengan permintaan konsumen. Ada rasa kemandirian lokasi bahwa pelanggan pada umumnya tidak memiliki kontrol atau pengetahuan atas keberadaan lokasi sumberdaya yang disediakan, tetapi ada kemungkinan dapat menentukan lokasi di tingkat yang lebih tinggi (misalnya, negara, negara bagian, atau

datacenter). Contoh sumberdaya termasuk penyimpanan, pemrosesan, memori, bandwidth jaringan, dan mesin virtual.

1. *Rapid elasticity*. Kemampuan dapat ditetapkan dan dirilis secara elastis, dalam beberapa kasus dilakukan secara otomatis untuk menghitung keluar dan masuk dengan cepat sesuai dengan permintaan. Untuk konsumen, kemampuan yang tersedia yang sering kali tidak terbatas dan kuantitasnya dapat disesuaikan setiap saat.
2. *Measured Service*. Sistem *cloud computing* secara otomatis mengawasi dan mengoptimalkan penggunaan sumber daya dengan memanfaatkan kemampuan pengukuran (*metering*) pada beberapa tingkat yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, *bandwidth*, dan *account* pengguna aktif). Penggunaan sumber daya dapat dipantau, dikendalikan, dan dilaporkan sebagai upaya memberikan transparansi bagi penyedia dan konsumen dari layanan yang digunakan[12].

2.2.13. Database Management System

Database Management System (DBMS) merupakan perangkat lunak untuk mengendalikan pembuatan, pemeliharaan, pengolahan, dan penggunaan data yang berskala besar. Penggunaan DBMS saat ini merupakan hal yang sangat penting dalam segala aspek, baik itu dalam skala yang besar atau kecil. Sebagai contoh media sosial facebook menggunakan DBMS untuk menyimpan data-data pengguna facebook yang sangat banyak kedalam DBMS MySQL.

Beberapa DBMS yang digunakan adalah MySQL dan MariaDB. Berdasarkan survey yang dilakukan, MySQL dan MariaDB merupakan DBMS yang banyak digunakan sebagai contoh survey yang terdapat pada *db-engines.com DB-Engines Ranking* menempatkan MySQL pada posisi ke-2 sedangkan MariaDB pada posisi ke-20 namun pada survey yang terdapat di *serverwatch.com Top 10 Enterprise Database System Of 2016*, MariaDB menempati posisi ke-6 dan MySQL menempati posisi ke-7[13].

2.2.14. MYSQL

MySQL merupakan *software database open source* yang paling populer di dunia. MySQL menjadi pilihan utama bagi banyak pengembang *software* dan aplikasi hal ini dikarenakan kelebihan MySQL diantaranya sintaksnya yang mudah dipahami, didukung program-program umum seperti C, C++, Java, PHP, Python. Pengguna MySQL tidak hanya sebatas pengguna seorang maupun perusahaan kecil, namun perusahaan seperti Yahoo!, Google, Nokia, Youtube, Wordpress juga menggunakan DBMS MySQL[13].

2.2.15. Python

Python merupakan bahasa pemrograman yang berorientasi objek dinamis, dimana dapat digunakan untuk bermacam-macam pengembangan perangkat lunak. Python disini media pendukung yang kuat untuk integrasi dengan bahasa pemrograman lain dan alat bantu lainnya. Python hadir dengan standar yang dapat diperluas serta dapat dipelajari hanya dalam beberapa hari. Sudah banyak programmer Python yang menyatakan bahwa mereka mendapatkan produktivitas yang lebih tinggi. Mereka juga merasakan bahwa Python meningkatkan kualitas pengembangan karena kode sumber yang mereka tulis dapat terus dipelihara.