

BAB I I

LANDASAN TEORI

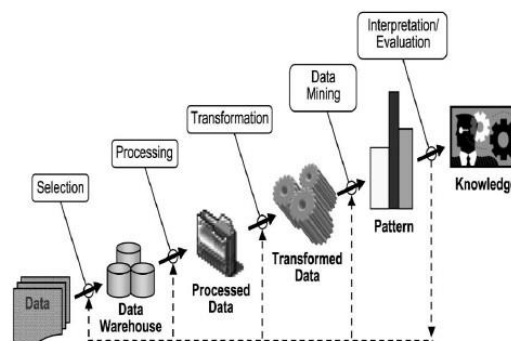
2.1 Data Mining

Data Mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. Data Mining adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidentifikasi Informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar .(Tarigan et al., 2022)

Data mining adalah proses menemukan pola, pengetahuan, atau informasi dalam kumpulan data yang sangat besar. Pengetahuan yang diperoleh dari proses ini dapat digunakan untuk membuat keputusan di masa depan.
(Adrianus & Lasut, 2023)

Data mining berkaitan dengan perencanaan integrasi dan penemuan ilmiah, interpretasi, dan visualisasi berbagai kumpulan data. Penemuan Pengetahuan dalam Database (KDD) adalah proses menyeluruh menemukan dan mengenali pola yang tidak sepele dalam data; pola yang ditemukan adalah valid, baru, bermanfaat, dan dapat dimengerti.
(Kusmawaty, 2017)

Berikut ini adalah tahapan dari Data Mining atau yang biasa disebut *Knowledge Discovery in Database* :



Gambar 2. 1. *Knowledge Discovery in Database*

2.1.1 Data Selection

1. Tetapkan himpunan data target, pilih himpunan data, atau fokus pada kumpulan variabel atau sampel data, tempat penemuan akan dibuat.
2. Pemilihan informasi dari sekumpulan data operasional harus diselesaikan sebelum tahap pemulihan data di KDD dimulai. Informasi yang dipilih untuk digunakan untuk proses penambangan data Informasi disimpan dalam catatan, terpisah dari database operasional.

2.1.2 *Pre-processing / Cleaning*

1. Pemrosesan awal dan pembersihan data adalah tugas dasar, misalnya penghapusan *noise* yang dilakukan
2. Sebelum proses data mining dapat diselesaikan, penting untuk melakukan siklus pembersihan pada Informasi yang menjadi titik fokus KDD.
3. Sistem pembersihan mencakup, antara lain, menghilangkan Informasi salinan, benar-benar memeriksa konsistensi Informasi, dan memperbaiki kesalahan dalam Informasi, seperti kesalahan pencetakan (tipografi).
4. Proses *enrichment*, merupakan proses “memperkaya” data yang ada dengan data atau Informasi lain yang relevan dan diolah untuk KDD, seperti data atau Informasi eksternal.

2.1.3 *Transformation*

1. Carilah sorotan yang berguna untuk mengumpulkan Informasi berdasarkan tujuan yang ingin Anda capai.
2. Merupakan proses perubahan pada data yang dipilih, dengan tujuan agar data tersebut sesuai untuk diproses data mining. Proses ini merupakan aliran inventif dan benar-benar bergantung pada jenis atau contoh data yang akan diperiksa dalam kumpulan data.

2.1.4 Data mining

1. Pemilihan tugas data mining, pemilihan goal dari proses KDD misalnya klasifikasi, regresi, *clustering*.
2. Memilih algoritma data mining untuk pencarian.
3. Proses data mining, yaitu mencari pola atau informasi menarik dalam data terpilih dengan menggunakan teknik atau metode tertentu. Teknik, metode atau algoritma dalam data mining sangat bervariasi, dan pemilihan metode atau algoritma yang tepat sangat bergantung pada tujuan dan proses KDD secara khusus.

2.1.5 Interpretation / Evaluation

1. Penerjemahan pola-pola yang dihasilkan dari data mining.
2. Pola informasi yang dihasilkan dari proses data mining harus ditampilkan dalam bentuk yang mudah dipahami oleh pihak yang berkepentingan.
3. Tahap ini merupakan bagian dari proses KDD dan mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta sebelumnya.

2.2 Klasifikasi

Dalam data mining terdapat beberapa teknik yang memiliki fungsi berbeda-beda yaitu prediksi, estimasi, deskripsi, klasifikasi, pengklasteran dan asosiasi. Klasifikasi merupakan metode yang digunakan untuk menemukan model atau fungsi yang digambarkan dengan perbedaan kelas data atau konsep yang berfungsi untuk memprediksi kelas dari objek yang label sudah diketahui.

klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk suatu model yang mampu membedakan data ke dalam kelas-kelas yang berbeda berdasarkan aturan atau fungsi tertentu.

Dalam konteks klasifikasi menggunakan *Naïve bayes*, kita mencoba memprediksi kelas dari suatu data baru berdasarkan atribut-atribut yang dimilikinya. Klasifikasi dilakukan dengan menghitung probabilitas untuk setiap kelas berdasarkan atribut data baru tersebut, dan kemudian memilih kelas dengan probabilitas tertinggi.

Proses Klasifikasi Menggunakan *Naïve bayes* dilakukan dengan beberapa langkah berikut :

- 2.2.1 Menghitung *Prior Probability*: Menghitung probabilitas dasar dari setiap kelas di dataset.
- 2.2.2 Menghitung *Likelihood*: Menghitung probabilitas terjadinya setiap atribut dalam data baru berdasarkan kelas yang diprediksi.
- 2.2.3 Menghitung Probabilitas Posterior: Menggabungkan *Prior Probability* dengan *Likelihood* menggunakan *Teorema bayes* untuk setiap kelas.
- 2.2.4 Menentukan Klasifikasi: Kelas dengan probabilitas tertinggi dianggap sebagai prediksi yang benar.

2.3 *Naïve bayes*

Naive Bayes adalah klasifikasi probabilitas dan statistik yang diciptakan oleh ilmuwan Inggris Thomas Bayes. Algoritma ini menggunakan pengalaman masa lalu untuk memprediksi peluang masa depan, yang juga dikenal sebagai Teorema Bayes, dan dikombinasikan dengan Naive Bayes, yang mengasumsikan kondisi antara atribut yang saling independen. Oleh karena itu, algoritma Naive Bayes menganggap bahwa apakah fitur tertentu dari satu kelas ada atau tidak ada hubungannya dengan fitur kelas lain.

Untuk memprediksi kemungkinan keanggotaan kelas, pengklasifikasi Bayesian, juga dikenal sebagai metode Bayesian naif, adalah teknik klasifikasi statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan kelas. Metode ini didasarkan pada teorema Bayes dan memiliki kemampuan untuk mengklasifikasikan data dengan cara yang mirip dengan jaringan syaraf dan pohon

keputusan. Metode ini terbukti akurat dan cepat ketika diterapkan pada database yang mengandung banyak data.(Linda et al., 2020)

Berikut ini adalah bentuk umum *Naïve bayes* sebagai berikut:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \dots \text{persamaan(1)}$$

Penjelasan dari persamaan sebagai berikut:

X : Data dengan *Class* yang belum diketahui

H : Hipotesis pada data X yang merupakan suatu *Class* khusus.

$P(H|X)$: Nilai *probabilitas* pada hipotesis H berdasarkan kondisi X.

$P(H)$: Nilai *probabilitas* pada hipotesis H.

$P(X|H)$: Nilai *probabilitas* X yang berdasarkan dengan kondisi H.

$P(X)$: Nilai *probabilitas* pada X.

Untuk menjelaskan *Naïve bayes*, perlu diketahui bahwa proses klasifikasi memerlukan instruksi tertentu untuk menentukan kelas yang sesuai untuk sampel yang dianalisis. Oleh karena itu, *Naïve bayes* diatas disesuaikan sebagai berikut:

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)} \dots \text{persamaan(2)}$$

Dalam persamaan (2) variabel C mewakili kelas, sedangkan $F_1 \dots F_n$ mewakili karakteristik petunjuk yang diperlukan untuk melakukan klasifikasi. Oleh karena itu, rumus menjelaskan bahwa probabilitas memasukkan sampel dengan karakteristik tertentu di kelas C (posterior) adalah probabilitas munculnya kelas C (Sebelum masuknya sampel, sering disebut sebelumnya), dikalikan dengan probabilitas munculnya karakteristik sampel di kelas C (juga disebut Kemungkinan), dibagi dengan probabilitas munculnya karakteristik sampel secara global (juga disebut bukti). Oleh karena itu, rumus di atas juga dapat ditulis secara sederhana sebagai berikut :

$$\text{Posterior} = \frac{\text{Prior} \times \text{likelihood}}{\text{evidence}} \dots \text{Persamaan(3)}$$

Nilai bukti di atas selalu tetap untuk setiap kelas dalam satu sampel. Nilai posterior dari sampel ini kemudian akan dibandingkan dengan nilai posterior dari sampel lain untuk menentukan kelas mana yang tepat untuk sampel tersebut.

2.4 Demam Berdarah Dengue (DBD)

Demam Berdarah Dengue (DBD) adalah penyakit menular yang disebabkan oleh infeksi virus dengue dan ditularkan melalui gigitan nyamuk *Aedes aegypti* dan *Aedes albopictus*. Virus dengue memiliki empat serotipe berbeda, yaitu DENV-1, DENV-2, DENV-3, dan DENV-4. Infeksi oleh salah satu serotipe tidak memberikan kekebalan terhadap serotipe lainnya, sehingga seseorang dapat terinfeksi lebih dari satu kali dalam hidupnya.

DBD ditandai dengan demam tinggi mendadak, nyeri otot, nyeri sendi, sakit kepala, ruam kulit, dan dalam kasus yang parah dapat menyebabkan perdarahan, kerusakan pembuluh darah, dan penurunan jumlah trombosit. Jika tidak ditangani dengan cepat dan tepat, DBD dapat menyebabkan syok dan kematian.

2.4.1 Penyebab

Penyebab utama DBD adalah virus dengue yang ditularkan oleh nyamuk betina dari spesies *Aedes aegypti* dan *Aedes albopictus*. Nyamuk ini umumnya menggigit pada pagi hari dan sore hingga petang hari, meskipun mereka juga dapat menggigit pada malam hari di area yang cukup terang.

Penularan virus dengue terjadi ketika nyamuk *Aedes* menggigit orang yang terinfeksi virus dengue, menghisap darah yang mengandung virus, dan kemudian menyebarkan virus tersebut ketika menggigit orang lain. Virus tidak menyebar langsung dari orang ke orang, sehingga kontak fisik tidak akan menyebabkan penularan.

2.4.2 Gejala

Gejala DBD biasanya muncul 4–10 hari setelah gigitan nyamuk yang terinfeksi. Gejala dapat bervariasi dari ringan hingga parah, dan beberapa di antaranya meliputi:

1. Demam tinggi mendadak biasanya mencapai 40°C atau lebih.
2. Sakit kepala parah terutama di bagian dahi.
3. Nyeri sendi dan otot dikenal sebagai "*breakbone fever*" karena rasa nyeri yang hebat.
4. Nyeri belakang mata rasa sakit di belakang bola mata yang sering kali diperburuk oleh gerakan.
5. Ruam kulit muncul beberapa hari setelah demam, berupa bintik merah yang tidak gatal.
6. Mual dan muntah sering kali disertai dengan hilangnya nafsu makan.

2.4.3 Tahapan Gejala

Penyakit DBD umumnya berkembang melalui tiga tahapan klinis:

1. Tahap Demam: Ditandai dengan demam tinggi, sakit kepala, nyeri sendi, dan gejala umum lainnya. Pada tahap ini, pemeriksaan darah mungkin menunjukkan penurunan trombosit yang ringan.
2. Tahap Kritis: Terjadi pada hari ketiga hingga ketujuh setelah demam. Pada tahap ini, demam mungkin mulai menurun, namun risiko perdarahan dan kebocoran plasma meningkat. Penurunan trombosit yang signifikan dan peningkatan hematokrit dapat terjadi, yang merupakan tanda risiko syok.
3. Tahap Pemulihan: Demam mulai mereda, kondisi umum pasien membaik, dan trombosit mulai meningkat. Namun, pemantauan masih diperlukan karena risiko komplikasi dapat terjadi.

2.5 Basis Data

Basis data adalah kumpulan Informasi yang diorganisasikan dan disimpan dalam komputer sehingga dapat diperiksa dengan program basis data. Produk yang digunakan untuk mengawasi dan memanggil pertanyaan administrasi kumpulan data dikenal sebagai kerangka kumpulan data (kerangka administrasi basis InFormasi, DBMS). Kerangka kumpulan data dapat dipelajari dalam ilmu data.

Konsep dasar dari basis data adalah kumpulan catatan, atau potongan Informasi. Basis data memiliki gambaran yang disebut pemetaan. Diagram yang menggambarkan item-item yang digambarkan dalam kumpulan data, hubungan antar artikel-artikel tersebut. Ada hubungan antara item-item ini. Ada banyak cara untuk memilah suatu konstruksi, atau memodelkan Desain Basis data atau Model Informasi. model yang umumnya digunakan saat ini adalah model revisional, yang dalam istilah awam menganggap semua data sebagai tabel yang saling berhubungan di mana setiap tabel terdiri dari garis dan segmen (definisi asli menggunakan susunan kata numerik). Dalam model ini, hubungan antar tabel dilihat menggunakan kualitas serupa antar tabel. model yang berbeda, misalnya model progresif dan Model organisasi menggunakan cara yang lebih tegas dalam menangani hubungan antar tabel.

Kumpulan data yang saling berhubungan disebut sebagai *database*, dan perangkat lunaknya disebut sebagai sistem manajemen *database* (DBMS). Dengan asumsi pengaturannya jelas, banyak administrator dan programmer menggunakan basisdata atau *database* untuk kedua arti tersebut. Dengan demikian, secara konsep basis data adalah kumpulan InFormasi yang menyusun catatan-catatan yang saling berhubungan (koneksi) dengan tujuan tertentu sehingga membentuk Informasi atau data baru. Atau, *database* adalah kumpulan data yang diorganisasikan menurut skema atau struktur tertentu dan dihubungkan (terkait) satu sama lain. Pada PC, kumpulan data disimpan dalam *hardware* berkapasitas peralatan, dan dengan pemrograman khusus yang dikontrol untuk tujuan atau penggunaan tertentu. Koneksi atau relasi Informasi biasanya merupakan kunci dari setiap rekaman saat ini. Informasi adalah realitas atau nilai (harga) yang terekam atau digambarkan gambaran suatu benda. Informasi yang dimanfaatkan merupakan kenyataan-kenyataan yang dicatat dan selanjutnya ditangani (ditangani) menjadi suatu struktur yang berharga atau bermanfaat bagi klien sehingga membentuk apa yang disebut dengan data. Jenis data yang rumit dan terkordinasi serta penanganan kumpulan data dengan PC akan dimanfaatkan untuk siklus dinamis dalam administrasi untuk membentuk kerangka data administrasi (MIS), InFormasi dalam kumpulan data adalah hal yang paling kecil

dan paling penting untuk membangun sebuah kumpulan data yang layak dan *valid*.
Data Basis Data terintegrasi dan dibagikan:

Terpadu (*integrated*) berkas-berkas data yang ada pada Basis Data saling terkait (terjadi despendensi data);

1. Berbagai data atau InFormasi serupa dapat dimanfaatkan oleh berbagai pengguna secara bersamaan. Sering disebut sebagai kerangka kerja multipengguna.

Data merupakan aset yang sangat berharga bagi hampir semua organisasi. Dengan tersedianya data yang melimpah, persoalan pengaturan data menjadi sesuatu yang vital dalam perbaikan kerangka data para eksekutif. Dengan cara ini, tujuan dari diadakan pengaturan data adalah sebagai berikut :

- 1) Memberikan kapasitas data atau Informasi untuk digunakan saat ini dan masa depan oleh organisasi.
- 2) Sebagai suatu pendekatan dalam memasukkan data atau Informasi dengan tujuan untuk mempermudah kesalahan administrator dan selanjutnya mencakup waktu yang diharapkan oleh pengguna untuk mendapatkan Informasi tersebut beserta kebebasan yang dimilikinya atas data yang dijaga.
- 3) Pengendalian data atau Informasi untuk setiap siklus sehingga Informasi terus diperbarui dan mencerminkan perubahan spesifik yang terjadi di setiap sistem.
- 4) Perlindungan data dari potensi peningkatan, penyesuaian, pencurian, dan pengaruh-pengaruh lainnya.

Suatu bangunan Basis Data memiliki jenjang sebagai berikut:

- 1) Karakter merupakan bagian terkecil dari data atau Informasi berupa angka, huruf, atau karakter unik yang menyusun suatu item atau *field*.
- 2) *Field*/item merupakan gambaran suatu kualitas dan rekaman (*record/tuple*) yang menunjukkan suatu item dari data.

- 3) *Records/Records/tuple* adalah bermacam-macam *field* yang menyusun suatu *record* atau rekaman. Catatan yang menggambarkan unit InFormasi individu tertentu.
- 4) Dokumen merupakan kumpulan *Records-Records* yang menggambarkan satu InFormasi atau data yang sejenis. Mengatasi setiap Informasi.
- 5) *Database* merupakan kumpulan catatan file atau tabel yang menyusun suatu basis Informasi yang membentuk *database*.

2.5.1 Normalisasi Basis Data

Menurut Normalisasi, ini adalah pengelompokan elemen data ke dalam tabel yang menunjukkan entitas dan hubungannya. Normalisasi juga merupakan proses pengelompokan atribut dan hubungan sehingga membentuk "hubungan struktur sumur", yang merupakan hubungan dengan sejumlah kecil kumpulan data (jumlah minimum redundansi), dan memberikan kemungkinan bagi Pengguna untuk memasukkan, menghapus, dan memodifikasi baris data dalam hubungan, yang tidak mengakibatkan kesalahan atau inkonsistensi data, disebabkan oleh operasi ini. Dalam proses normalisasi, perlu diketahui pengertian tahapan atau bentuk normalisasi, yaitu.

1. Tidak Normal (*Unnormalize*)

Bentuk normal adalah kumpulan data yang dicatat dan tidak mengikuti aturan atau format tertentu. Dalam bentuk abnormal terdapat kelompok berulang sehingga dalam kondisi seperti ini akan menimbulkan masalah dalam memanipulasi data seperti menyisipkan, memperbarui dan menghapus anomali. Anomali sisipan terjadi saat menambahkan data tetapi masih ada elemen yang masih kosong. Anomali pembaruan terjadi ketika ada perubahan dalam sejumlah data yang tidak digunakan dalam tabel tetapi tidak semua data diubah. Anomali penghapusan terjadi ketika baris atau rekaman yang tidak digunakan dihapus dan mengakibatkan hilangnya data lain.

Contoh: Bentuk tidak normal

Tabel 2. 1 Bentuk Tidak Normal

No. Faktur	Tanggal	Kode Pelanggan	Nama	Kode Barang	Nama Barang	Harga	Qty
A-101	08/12/2023	R-112	Jora	J-010	Kabel	102.000,-	2
				J-100	Handuk	50.000,-	4
B-102	13/12/2016	R-113	Ratu	J-090	Tali	5.000,-	5
				J-132	Teh	23.000,-	1

2. Normalisasi Pertama

Dalam hubungan database, grup berulang tidak boleh digunakan karena dapat menyebabkan anomali. Oleh karena itu, bentuk normal pertama akan dihasilkan dari tahap unnormalisasi. Dalam normalisasi pertama, relasi atau tabel harus memenuhi normal pertama jika dan hanya jika setiap atribut relasi hanya memiliki satu nilai dalam baris atau rekaman. Dalam normalisasi pertama, data dibentuk dalam satu catatan, dan nilai bidang adalah nilai yang tidak dapat dibagi.

Contoh: Bentuk normalisasi pertama (1 NF)

Berikut perubahannya:

Tabel 2. 2 Bentuk Normalisasi Pertama (1 NF)

No. Faktur	Tanggal	Kode Pelanggan	Nama	Kode Barang	Nama Barang	Harga	Qty
A-222	21/08/2024	P-001	M Fikri Setiadi	B-001	Sampo	12.000,-	1
A-222	21/08/2024	P-001	M Fikri Setiadi	B-002	Kopi	15.000,-	1
A-333	21/08/2024	P-002	Jack	B-002	Kopi	15.000,-	1
A-333	21/08/2024	P-002	Jack	B-003	The	7.000	2

3. Normalisasi Kedua

Dalam merancang hubungan Database, dependensi fungsi parsial ke kunci utama tidak diperbolehkan, karena dapat berdampak pada anomali. Oleh karena itu tahap normalisasi pertama akan menghasilkan bentuk normal kedua yang dapat didefinisikan sebagai berikut

Contoh: Bentuk normalisasi kedua (2 NF)

Bentuk perubahannya:

Tabel 2. 3 Bentuk Normalisasi Kedua (2 NF)

No. Faktur	Tanggal	Kode Pelanggan	Kode Barang	Qty
A-111	08/12/2023	C-111	D-111	2
A-111	08/12/2023	C-111	D-111	2
A-222	08/12/2023	C-222	D-111	2
A-222	08/12/2023	C-222	D-111	3

Kode Pelanggan	Nama
A-111	Calvin
A-222	Nahum

Kode Barang	Nama Barang	Harga
B-111	Handuk	15.000,-
B-111	Meja	17.000,-
B-111	Kursi	15.000,-

4. Normalisasi Ketiga

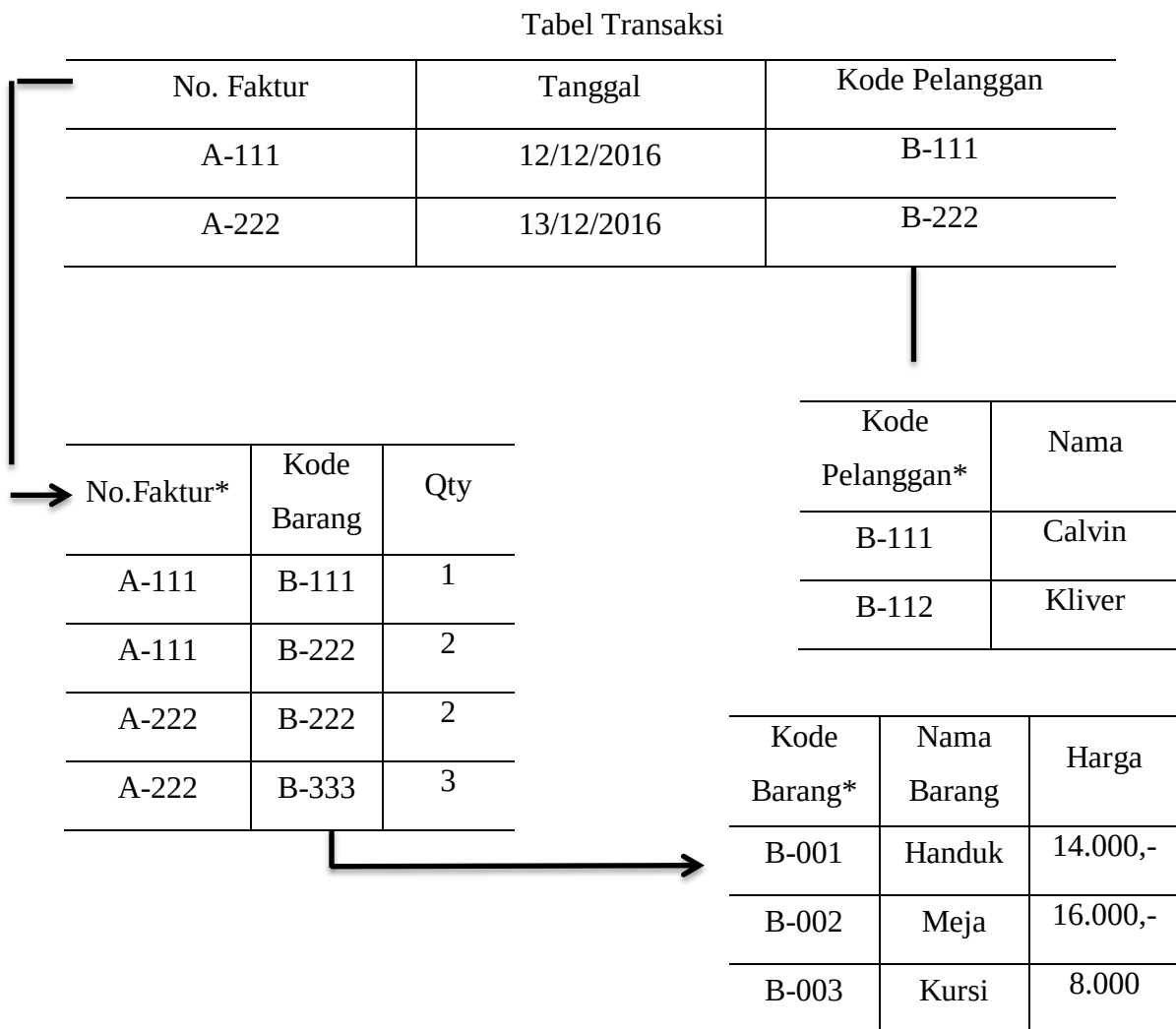
Dalam desain hubungan Database, dependensi transitif tidak diperbolehkan karena dapat berdampak pada terjadinya anomali. Untuk alasan ini, tahap ketiga normalisasi harus dibentuk yang dapat didefinisikan sebagai berikut:

Sebuah hubungan dapat dikatakan sebagai normalisasi ketiga jika dan hanya jika hubungan tersebut memenuhi normal kedua dan atribut non-kunci apa pun tidak memiliki dependensi fungsional transitif ke kunci utama. Ini berarti bahwa setiap atribut yang bukan kunci harus bergantung hanya pada kunci utama secara keseluruhan dan bentuk normalisasi ketiga telah diperoleh dalam tabel yang optimal.

Contoh: Bentuk normalisasi ketiga (3 NF)

Bentuk perubahannya:

Tabel 2. 4 Bentuk Normalisasi Ketiga (3 NF)



Ada beberapa kunci yang digunakan untuk proses pencarian, pemfilteran, dan penghapusan yang biasa digunakan dalam pengelompokan database, yaitu:

1. Kunci Super (Kunci Super) Kumpulan satu atau beberapa entitas yang dapat digunakan untuk mengidentifikasi entitas secara unik dalam kumpulan entitas.
2. Kunci Utama Atribut atau sekumpulan atribut minimal yang tidak hanya secara unik mengidentifikasi peristiwa tertentu, tetapi juga dapat mewakili setiap peristiwa entitas.
3. Kunci Asing Atribut atau kumpulan atribut yang melengkapi hubungan yang menunjukkan kepada induk.
4. Kunci Kandidat Atribut atau kumpulan atribut minimum yang secara unik mengidentifikasi peristiwa tertentu dari suatu entitas.
5. Kunci Alternatif Kunci kandidat yang tidak digunakan sebagai kunci utama, seringkali kunci alternatif ini digunakan sebagai kunci pengurutan untuk membuat laporan.
6. Kunci Gabungan atau Kunci Campuran, atau *Composite Key*, dapat digunakan untuk menggabungkan beberapa baris menjadi satu jika baris tidak dapat digunakan sebagai kunci utama.

Bahasa database adalah kumpulan perintah, juga dikenal sebagai pernyataan, yang dapat dibuat dan diberikan oleh pengguna, dan dikenali atau diproses oleh DBMS untuk melakukan tindakan tertentu.

Bahasa Basis Data biasanya dibagi menjadi dua kategori:

2.5.2 Data Definition Language (DDL)

Bahasa Definisi Data (DDL)—Struktur atau skema database menjelaskan dan mewakili desain database secara keseluruhan. Dengan menggunakan bahasa ini, kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dan lainnya.

2.5.3 Data Manipulation Language (DML)

Bahasa Manipulasi Data (DML) adalah jenis bahasa Basis Data yang bermanfaat untuk melakukan manipulasi dan pengambilan data pada suatu Basis Data. Jenis manipulasi data mencakup menambahkan atau mengubah data baru ke Basis Data, menghapus data dari Basis Data, dan mengubah data.

- 1) Menambah atau mengubah data baru ke Basis Data;
- 2) Menghapus data
- 3) Mengubah data di Basis Data

2.6 MySQL

MySQL adalah bahasa yang digunakan untuk mendapatkan informasi dalam database. Ini juga merupakan salah satu aplikasi populer dalam kemajuan web server Apache dan bahasa pemrograman PHP, dan merupakan pelaksana kerangka administrasi kumpulan data sosial (RDBMS) yang tersebar secara bebas. Selain itu, MySQL memiliki kemampuan untuk menjalankan perintah Bahasa Pertanyaan Organisasi (SQL) untuk menangani kumpulan data yang dimilikinya. Saat ini, trigger didukung oleh MySQL, yang membuat pengelolaan tabel database lebih mudah. (Gusman & S, 2022).

MySQL adalah bahasa pemrograman kumpulan data sumber terbuka yang banyak digunakan untuk menangani Informasi penting menggunakan bahasa SQL. MySQL juga disebut SQL, yang merupakan kependekan dari Bahasa Pertanyaan Terorganisir. SQL adalah bahasa terorganisir yang digunakan untuk menangani kumpulan data. MySQL adalah kerangka administrasi kumpulan data sosial. Artinya, pengawasan Informasi dalam kumpulan data akan ditempatkan dalam beberapa tabel terpisah sehingga pengendalian Informasi akan jauh lebih cepat. MySQL dapat digunakan untuk menangani basis Informasi mulai dari yang kecil hingga yang sangat besar.

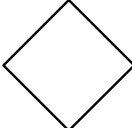
SQL juga dapat diartikan sebagai standar titik koneksi untuk kerangka administrasi sosial. sistem yang berjalan pada komputer pribadi disertakan. SQL

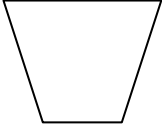
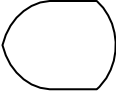
memungkinkan klien mengetahui di mana sesuatu ditemukan, atau cara data diatur. SQL lebih muda digunakan dan dikenali dari lembar perhitungan dan dialek pemrograman penanganan Informasi. Penjelasan dasar SQL dapat menyebabkan serangkaian permintaan data yang disimpan di berbagai PC di lokasi yang tersebar, sehingga memerlukan banyak investasi dan sumber daya pemrosesan.

2.7 Flowchart

Flowchart merupakan gambaran aliran Informasi yang ditempatkan pada kerangka strategi yang mencirikan perencanaan aliran data ke dalam struktur program yang digunakan untuk membantu klien dalam mengembangkan kerangka tersebut. Kemampuan grafik adalah untuk menggambarkan dan membingkai suatu perkembangan sehingga terlihat jelas dan jelas dalam melihat rangkaian langkah-langkah pada saat yang sama. Gambar yang biasa digunakan dalam kerangka harus terlihat di tabel. (Amri et al., 2020)

Tabel 2. 5 *Flowchart*

Simbol	Keterangan
	Proses atau proses: menunjukkan tindakan atau proses yang sedang dilakukan.
	I/O adalah istilah yang mengacu pada proses input dan output data, terlepas dari jenis perangkat kerasnya.
	Terminal atau akhir proses menunjukkan awal atau akhir proses.
	Solusi: Pilihan kondisi program terdiri dari dua kemungkinan jawaban.

	Gambar <i>flowline</i>/saat ini: menunjukkan alur suatu proses.
	Dokumen/document: menunjukkan dokumen input dan output baik yang dilakukan secara manual atau melalui printer. Manual:
	Manual: menunjukkan pekerjaan yang dilakukan secara manual atau yang dilakukan secara komputer.
	Display: menampilkan output di monitor..
	Keyboard: menggunakan keyboard online untuk memasukkan data secara manual.
	Tempat penyimpanan disk: menunjukkan input dan output yang menggunakan disk magnetik. adalah file data/database.
	Penghubung/penghubung: menunjukkan hubungan antara proses-proses yang tidak terlibat atau terputus yang tetap berada di halaman yang sama.
	Penghubung halaman lain atau konektor juga dapat digunakan untuk menunjukkan hubungan ini.

2.8 PHP Dan Fremwork Laravel

PHP adalah bahasa scripting yang ditempatkan di server dan diproses oleh server. Selain itu, PHP adalah salah satu dari beberapa bahasa pemrograman HTML yang dibuat oleh Rasmus Ledorf, yang dimulai sebagai proyek pribadi dan dimodifikasi menjadi PHP oleh sekelompok enam programmer.(Ruseno, 2019)

PHP adalah singkatan dari PHP Hypertext Preprocessor yang merupakan bahasa skrip yang ditempatkan di server. Dan diproses di server. Hasilnya dikirim ke klien tempat pengguna menggunakan browser. PHP dirancang khusus untuk membuat aplikasi web dinamis, yang berarti dapat menghasilkan layar sesuai permintaan saat ini.(Utomo & Hamdani, 2021)

2.9 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa standar yang banyak digunakan untuk mendefinisikan prasyarat, membedah dan merencanakan, serta memahami objek-objek yang terletak di bidang teknik pemrograman di dunia modern. UML adalah bahasa tampilan PC dan korespondensi visual yang memanfaatkan grafik dan dukungan teks. Oleh karena itu, pemanfaatan UML tidak terbatas pada satu teknik saja, melainkan UML umumnya digunakan dalam filosofi yang terletak pada objek.(Rambe et al., 2020)

Struktur dan pendekatan peModelan desain program berorientasi objek (OOP) dan penerapannya adalah *Unified Modeling Language (UML)*. UML adalah strategi untuk mengembangkan kerangka kerja tersebut.(Pakaya et al., 2020)

2.9.1 *Use case Diagram*

Use case diagram merupakan Model yang menggambarkan cara kerja *framework* yang akan dibuat. *Use case diagram* menggambarkan kerja sama setidaknya satu penghibur (Aktor) dengan kerangka kerja (Sistem). Tujuan dari *Use case diagram* adalah untuk mengidentifikasi fungsi-fungsi sistem dan siapa yang mempunyai izin untuk menggunakannya. Sederhananya, *Use case* menampilkan sistem dari sudut pandang pengguna.(Melanda et al., 2023)

Ada 2 elemen penting yang harus digambarkan yaitu:

1. Aktor

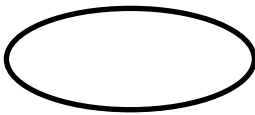
Aktor adalah apa pun yang bekerja sama langsung dengan sistem, bisa saja itu adalah pengguna (yang ditunjukkan oleh pekerjaannya dan bukan nama/tenaga kerja atau sistem PC lainnya. *The stick-man*, gambar seseorang dengan kata benda di bagian bawah, digunakan untuk mengidentifikasi pelaku, menyatakan pekerjaan/kerangka kerja.

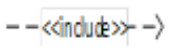
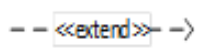
2. *Use case*

Use case ditandai dengan gambar oval dengan nama kata tindakan dinamis di dalamnya yang mengekspresikan gerakan sesuai sudut pandang penghibur. Setiap aktor dimungkinkan untuk berkomunikasi dengan kerangka kerja dalam banyak *Use case*. Selain itu, setiap kasus pemanfaatan dapat dilakukan oleh lebih dari satu aktor.

Berikut symbol-simbol dalam *Use case* diagram:

Tabel 2. 6 simbol-simbol dalam *Use case* diagram

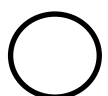
Simbol	Keterangan
	Aktor adalah apa pun yang berhubungan langsung dengan sistem, cenderung individu (yang muncul atas permintaan dan bukan nama) atau kerangka PC lain.
	<i>Use case</i> ditandai dengan gambar oval dengan nama kata tindakan dinamis di dalamnya yang mengekspresikan gerakan sesuai sudut pandang aktor.
	<i>Association</i> menggunakan panah terbuka untuk menunjukkan kapan aktor berinteraksi secara pasif dengan sistem.

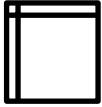
	<i>Generalization</i> yang digambarkan dengan garis tanpa baut membedakan siapa atau apa yang menyebut komunikasi secara langsung dan bukannya mengidentifikasi data.
	Termasuk memanggil fungsi program atau pemanggilan sebuah fungsi program (wajib) lainnya.
	<i>Extend</i> , merupakan perluasan dari <i>Use case</i> lain jika kondisi atau syarat tertentu.

2.9.2 Activity diagram

Activity diagram adalah bagan yang menggambarkan proses kerja atau latihan suatu sistem atau proses bisnis. Diagram aktivitas menggunakan simbol-simbol berikut:

Tabel 2. 7 Simbol- Simbol *Activity* Diagram

Simbol	Keterangan
	Titik Awal : Letaknya di pojok kiri atas dan merupakan awal dari aktivitas
	Titik Akhir: Diletakan pada akhir aktivitas
	Aktivitas: Menggambarkan interaksi atau pergerakan bisnis
	keputusan: Menyebarkan dimana terdapat lebih dari satu keputusan pergerakan

	Gabung: Penggabungan di mana lebih dari satu tindakan digabungkan menjadi satu
	<i>Swimlane</i> : Organisasi bisnis terpisah yang bertanggung jawab atas aktivitas yang terjadi

2.9.3 Sequence diagram

Sequence diagram adalah jenis UML yang menggambarkan komunikasi antara objek di dalam dan di sekitar sistem kerja, termasuk klien, tampilan, dan beberapa di antaranya adalah pesan yang digambarkan setelah beberapa waktu.

Tabel 2. 8 Simbol-Simbol *Sequence diagram*

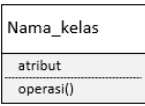
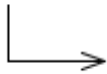
Simbol	Keterangan
Aktor 	<i>Actor Lifeline</i> , yaitu entertainer yang menggambarkan objek di luar sistem atau bukan sekadar pengguna sistem yang bekerja sama dengan sistem yang sedang dibangun atau dibuat dan <i>life saver</i> menyatakan keberadaan suatu objek.
Garis hidup/ <i>lifeline</i> 	<i>Object</i> , menyatakan objek yang berinteraksi dengan pesan.

Waktu aktif 	<i>Activation</i>, menunjukan objek dalam keadaan berfungsi dan berinteraksi, segala sesuatu yang berhubungan dengan waktu yang aktif ini merupakan sebuah tahapan yang dilakukan. Membahas pelaksanaan aktivitas suatu item, panjang ini langsung relatif terhadap waktu berlakunya atau rentang suatu aktivita
Pesan tipe <i>call</i> 	<i>Self Message</i>, menyatakan bahwa suatu artikel menyebut suatu aktivitas atau strategi yang ada pada item lain atau dirinya sendiri
Pesan tipe <i>send</i> 	<i>Message</i>, menyatakan bahwa suatu objek mengkomunikasikan sesuatu yang spesifik sebagai InFormasi atau data kepada objek lain dengan arah baut yang menunjukkan judul objek tujuannya.
Pesan tipe <i>return</i> 	<i>Return Message</i>, Menyatakan bahwa suatu item yang telah menjalankan suatu kemampuan atau strategi akan menyampaikan pesan kembali ke item tertentu, jalannya berfokus pada artikel yang menerima pesan tersebut kembali

2.9.4 Class diagram

Diagram kelas menggambarkan struktur platform sejauh itu mencirikan kelas-kelas yang akan dibuat untuk membangun platform. Kelas memiliki apa yang disebut atribut dan strategi atau tugas. Atribut adalah faktor yang diperlukan untuk kelas, dan teknik atau tugas adalah kemampuan yang dimiliki kelas. Kelas-kelas dalam struktur framework harus dapat memenuhi peran sesuai dengan prasyarat framework, sehingga pembuat perangkat lunak dapat membuat kelas-kelas dalam perangkat lunak sesuai dengan rancangan diagram kelas.

Tabel 2. 9 Simbol-Simbol *Class* diagram

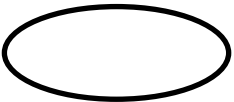
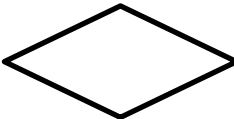
Simbol	Keterangan
Kelas 	Kelas pada struktur sistem.
Asosiasi 	Relasi antar kelas dengan makna umum.
Asosiasi berarah/<i>Directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi.
Kebergantungan/<i>Dependency</i> 	Kebergantungan antar kelas.
Agresi/<i>Aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.10 Entity Relationship Diagram (ERD)

Diagram struktural yang disebut diagram hubungan entitas, atau ERD, digunakan untuk mendesain *database*. ERD menggambarkan Informasi yang akan dimasukkan ke dalam kerangka dan batasannya. Bagian mendasar yang diingat untuk ERD adalah himpunan entitas, himpunan relasi, dan batasan. (Larassati et al., 2022)

Menurut (Togatorop et al., 2021) Entity *relationship* diagram akan menggambarkan Informasi yang disimpan dalam kerangka dan batasannya. ERD memiliki tiga gagasan pokok, yaitu:

Tabel 2. 10 Simbol-Simbol *Entity relationship* diagram

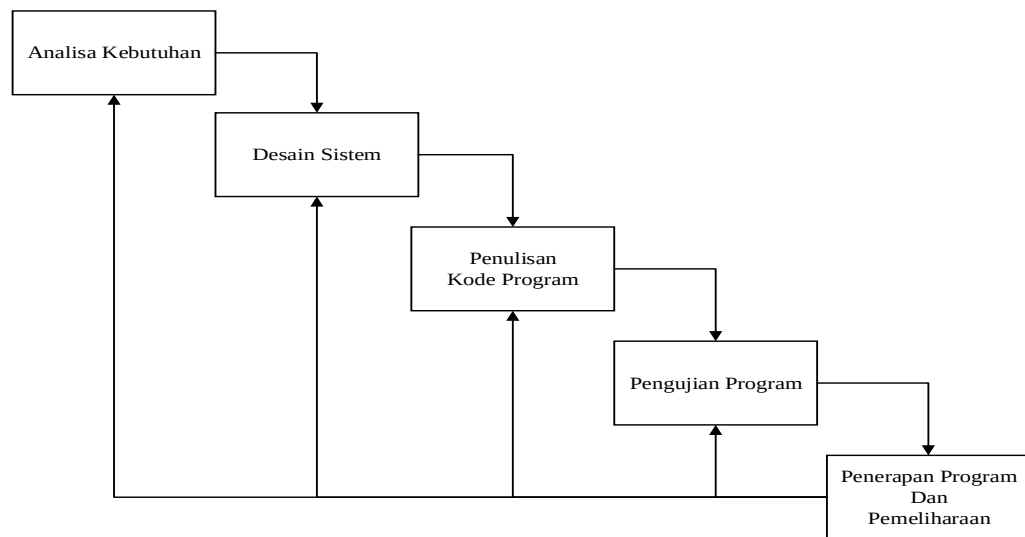
Simbol	Nama	Keterangan
	Entitas	Suatu entitas dapat berupa individu, tempat, objek, atau peristiwa yang dapat dipandang penting bagi suatu asosiasi atau organisasi. Setiap entitas memiliki beberapa atribut yang pada elemen tersebut harus disingkirkan dan disimpan dalam Kumpulan Data
	Atribut	Setiap entitas mempunyai kualitas tertentu yang disebut atribut. Mengatribusikan kemampuan untuk menggambarkan atribut yang disimpan dalam kumpulan data
	Relasi	Hubungan adalah hubungan antara setidaknya dua entitas yang berkaitan. Koneksi dalam ERD dapat dipahami dengan menggunakan gambar belah ketupat (permata).

2.11 Model Pengembangan Sistem

2.11.1 Model *Waterfall*

Metode ini juga bisa disebut dengan *Classic life cycle* atau *linear sequential Model*. Metode ini membutuhkan pendekatan sistematis dan sekuensial dalam mengembangkan perangkat lunak. Pemodelan ini menyangkut aktivitas berikut:

1. Merancang dan menampilkan kerangka/data (*Framework/Data Designing and Demonstrating*) Tahap ini kadang juga disebut *project definition*. Latihan dimulai dari pengembangan dan penentuan kebutuhan seluruh komponen kerangka kerja. Hal ini penting, karena pemrograman harus berhubungan dengan peralatan, individu, dan Informasi tersembunyi.
2. Analisis kebutuhan perangkat lunak (*Software Requirements Analysis*) Analisis kebutuhan dilakukan dengan mencari tahu dan mengumpulkan fungsi-fungsi perangkat lunak yang dibutuhkan. Hasilnya harus didokumentasikan dan *di-review* ke pelanggan.
3. Desain (*Design*) Proses desain menjelaskan kebutuhan-kebutuhan menjadi bentuk gambaran aktivitas yang akan terjadi, tampilan antarmuka, arsitektur perangkat lunak dan struktur data yang dimengerti perangkat lunak sebelum penulisan program dimulai.
4. Penulisan Program (*Coding*) Penulisan program dilakukan dengan mengubah desain menjadi bentuk yang dimengerti dan dipahami oleh mesin atau komputer dengan menggunakan bahasa pemrograman dan menghasilkan perangkat lunak yang sesuai dengan kebutuhan
5. Pengujian dapat dimulai setelah kode program ditulis dan program dapat dijalankan. Pengujian berpusat pada alasan internal dari perangkat lunak, kemampuan eksternal, mencari potensi kesalahan, dan ingin menjamin hasilnya.
6. Pemeliharaan (*Support/Maintenance*) Setelah perangkat lunak diberikan pada pelanggan, mungkin dapat ditemui *error* ketika dijalankan di lingkungan pelanggan. Pemeliharaan ini dapat mempengaruhi semua tahap yang dilakukan sebelumnya.

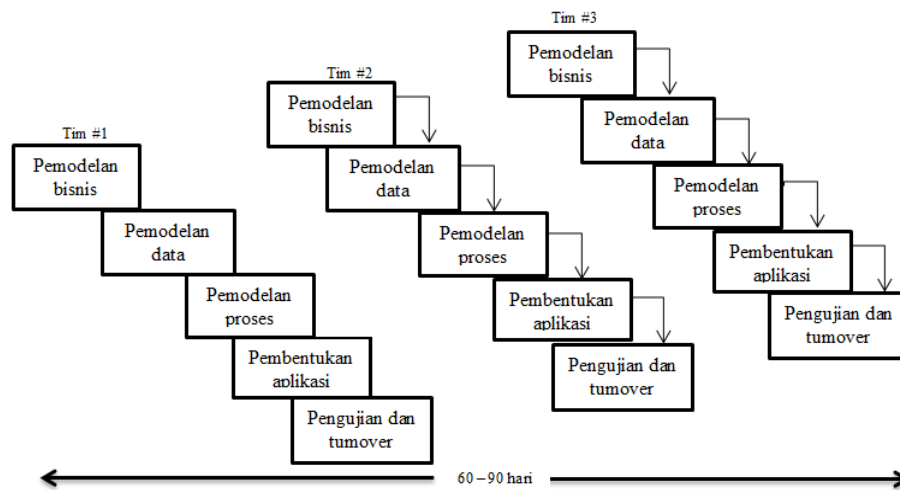


Gambar 2. 2 Metode *Waterfall*

2.11.2 Model *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah Model proses pengembangan perangkat lunak yang siklus kemajuannya sangat singkat (kira-kira 60 hari hingga 90 hari). Model RAD ini adalah versi "kecepatan tinggi" dari Model sekuensial linier yang memungkinkan konstruksi berbasis komponen untuk pengembangan.

Menurut, RAD adalah cara yang terlibat untuk menangani kerangka perbaikan yang mencakup strategi perbaikan dan perangkat pemrograman. RAD berharap untuk mempersingkat waktu yang biasanya diharapkan dalam siklus hidup pengembangan kerangka konvensional antara perencanaan dan pelaksanaan kerangka data. Pada akhirnya, RAD juga berupaya memenuhi kebutuhan bisnis yang berubah dengan cepat.



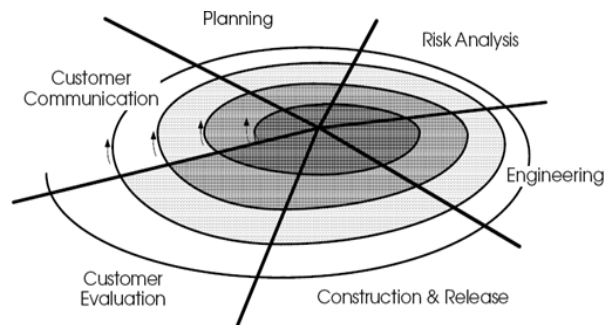
Gambar 2. 3 Model Rapid Application Development (RAD)

2.11.3 Model Spiral

Model spiral adalah Model siklus produk yang menggabungkan jenis campuran Model *prototipe* dengan kontrol dan bagian presisi dari Model berturut-turut lurus, dengan perluasan komponen lain, khususnya pemeriksaan resiko. Model ini memiliki beberapa latihan penting, yaitu:

1. Perancangan (*Planning*), penentuan tujuan, alternatif, dan batasan.
2. Analisis resiko (*Risk Analysis*), analisis alternatif, dan identifikasi/pemecahan resiko.
3. Rekayasa (*engineering*), pengembangan level berikutnya dari produk.

Evaluasi pemakai (*costumer evaluation*), penilaian terhadap hasil rekayasa bentuk spiral memberikan gambaran bahwa semakin besar iterasinya, maka menunjukkan makin lengkap versi dari perangkat lunak yang dibuat. Selama awal sirkuit, objektif, alternatif, dan batasan didefinisikan serta resiko diidentifikasi dan dianalisa.

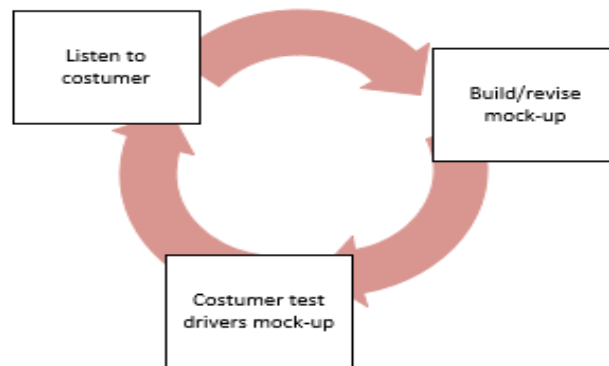


Gambar 2. 4 Model spiral

2.11.4 Model *Prototype*

Model (*prototyping* Model) dimulai dari pengumpulan kebutuhan klien terhadap perangkat yang akan dibuat. Kemudian program Model dibuat sehingga klien dapat membayangkan apa yang benar-benar mereka bayangkan. Biasanya, program *Prototype* tidak lengkap. Biasanya program ini menampilkan simulasi perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Model program ini dinilai oleh klien atau *User* sampai dilacak penentuan yang memenuhi keinginan klien atau keinginan *User*.

Berikut ini gambar dari *Model Prototype* :

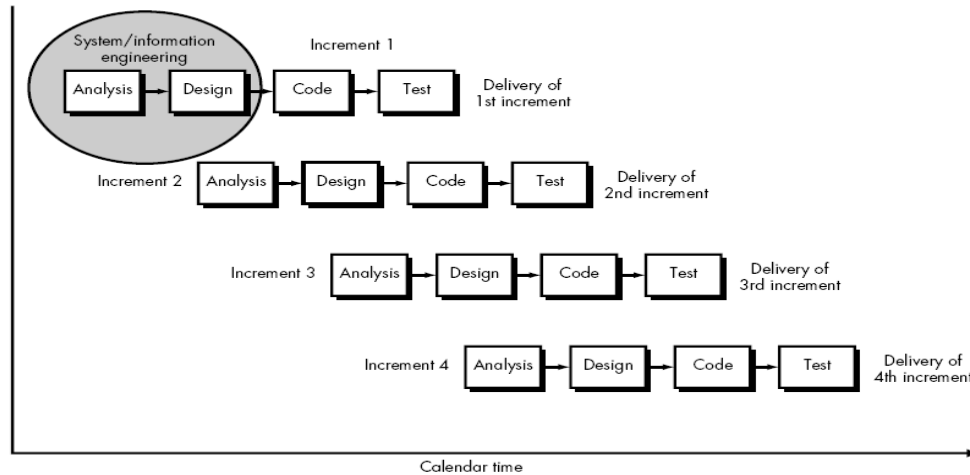


Gambar 2. 5 Model *Prototype*

2.11.5 Model *Iteratif*

Proses dalam Model air terjun dan proses *Iteratif* dalam Model *prototipe* digabungkan dalam Model *Iteratif*, disebut juga Model berulang. Model mantap akan

menghadirkan varian pemrograman yang memiliki kemampuan ekstra pada setiap pengembangan (peningkatan). Berikut gambar Model bertahapnya.



Gambar 2. 6 Model *Iteratif*

Model incremental dibuat untuk mengatasi kekurangan Model air terjun yang tidak mewajibkan adanya siklus, dan kekurangan Model strategi yang mempunyai interaksi yang terlalu singkat dan setiap penekanan interaksi belum tentu menciptakan suatu item (harus sebuah contoh). Model bertahap memberikan item atau aplikasi untuk setiap tahap tetap.

2.12 Pengujian *Black box*

Pengujian *black box* merupakan metode pengujian produk yang berfokus pada perbedaan elemen luar dari produk yang dibuat. Pengujian *black box* pada umumnya akan mengenali banyak hal, misalnya kemampuan yang salah atau hilang, kesalahan kumpulan data, kesalahan struktur Informasi, kesalahan akses Informasi, kesalahan antarmuka, kesalahan struktur Informasi, dan sebagainya. kesalahan kinerja, inisialisasi, dan amplifikasi, serta kesalahan pengguna *Black box* adalah teknik pengujian produk yang tidak menekankan seluk-beluk.