

BAB 2

LANDASAN TEORI

1.1. Pengertian Aplikasi

Menurut Buyens [9] aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu language software, yang dapat berbentuk assembler, compiler ataupun interpreter. Jadi language software merupakan bahasanya dan program yang ditulis merupakan program aplikasinya. Language software berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Bila hendak mengembangkan suatu program aplikasi untuk memecahkan permasalahan yang besar dan rumit, maka supaya program aplikasi tersebut dapat berhasil dengan baik, maka dibutuhkan prosedur dan perencanaan yang baik dalam mengembangkannya.

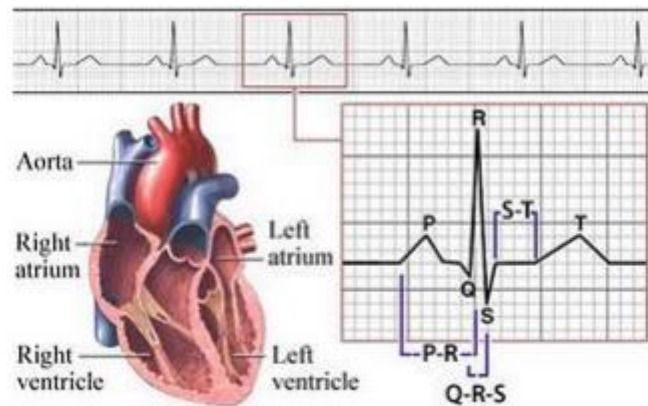
Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi internasional. Program-program paket tersebut dapat diandalkan, dapat memenuhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, user friendly (mudah digunakan), mempunyai dokumentasi manual yang memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya. Akan tetapi, bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri.

1.2. Pengertian Aplikasi Mobile

Mobile diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon mobile berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Menurut Buyens [9] aplikasi mobile berasal dari kata application dan mobile. Application yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan mobile dapat diartikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Maka aplikasi mobile dapat diartikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi mobile ini dapat diakses melalui perangkat nirkabel, pager, PDA, telepon seluler, smartphone, dan perangkat sejenisnya.

1.3. Pengertian Elektrokardiogram(EKG)

Elektrokardiografi (EKG) atau Electrocardiography (ECG) merupakan suatu alat yang digunakan untuk merekam sinyal biologi yang terbentuk sebagai hasil dari aktivitas listrik jantung. Sinyal ini diambil dengan cara memasang elektroda pada titik tertentu pada bagian tubuh pasien. Hasil rekaman EKG mempunyai bentuk yang spesifik sehingga dapat dijadikan sebagai acuan untuk menentukan kondisi kesehatan jantung seseorang oleh dokter ahli jantung. Sinyal EKG mempunyai tegangan sampai 0,3mV dan rentang frekuensi antara 0,03 – 100 Hz. Sinyal ini dideteksi dan direkam menggunakan perangkat elektrokardiografi. Pada dasarnya EKG terdiri dari banyak gelombang, yang tiap gelombang mewakili satu denyut jantung (satu kali aktifitas listrik jantung). Lihat gambar 3 gelombang EKG.



Sumber gambar : The Respiratory System and the Cardiovascular System.[13]

Gambar 1 Gelombang sinyal EKG

Dalam satu gelombang EKG terdiri dari beberapa titik gelombang ada yang disebut interval dan segmen. Titik terdiri dari titik P, Q, R, S, T dan U (kadang sebagian referensi tidak menampilkan titik U) sedangkan Interval terdiri dari PR interval, QRS interval dan QT interval dan Segmen terdiri dari PR segmen, dan ST segmen.

Sumber Tabel : The Respiratory System and the Cardiovascular System[13]

Tabel 1 Parameter Elektrokardiografi

Gelombang EKG	Amplitudo	EKG Interval	Durasi
P	<0.3 mV	P-R	0.12-0.20 dtk
R	1.6 – 3 mV	Q-T	0.35-0.44 dtk
Q	25% dari R	S-T	0.05-0.15 dtk
T	0.1 – 0.5 mV	Q-R-S	0.06 – 0.10 dtk

Interval antara R-R menandakan periode dari detak jantung yang dapat dikonversikan menjadi Heart Rate:

$$HR = 60 / R-R (S)$$

R – R = adalah interval antara sinyal R dengan sinyal R yang diukur dalam detik. Interval R-R relatif konstan dari detak ke detak. Perubahan pada interval R-R menandakan adanya kecepatan jantung yang tidak wajar. Dalam pengambilan sinyal elektrokardiografi terdapat berbagai metode yang bisa dilakukan yaitu :

1. *Standard Clinical EKG*

Menggunakan 10 elektroda (12 lead) digunakan untuk menganalisis kondisi kesehatan jantung pasien.

2. *Vectorcardiogram/ Standart Monitoring*

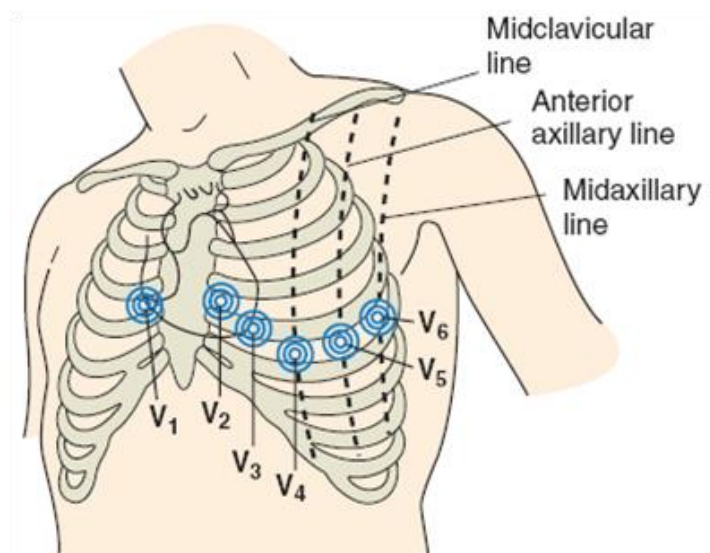
Pemodelan potensial tubuh sebagai vektor 3 dimensi dengan menggunakan sadapan bipolar (Einthoven) atau Unipolar. Pengambilan sinyal jantung melalui 3 titik tertentu pada tubuh, yang digunakan untuk memantau kondisi kesehatan jantung pasien dalam jangka waktu tertentu.

1.4. Sadapan (lokasi penempatan lead) EKG

Untuk memperoleh rekaman EKG dipasang elektroda-elektroda di kulit pada tempat-tempat tertentu. Lokasi penempatan elektroda sangat penting diperhatikan, karena penempatan yang salah akan menghasilkan pencatatan yang berbeda. Terdapat 3 jenis sadapan (lead) pada EKG, yaitu :

A. Sadapan Prekordial

Merupakan sadapan V1, V2, V3, V4, V5, dan V6 yang ditempatkan secara langsung di dada.



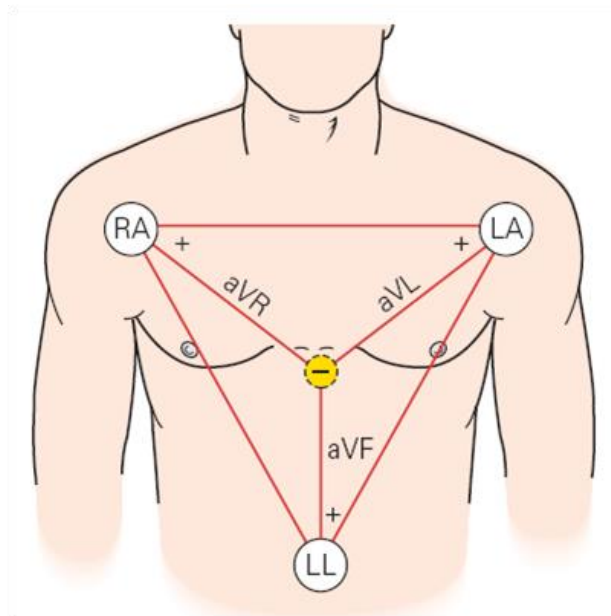
Sumber gambar : *The Respiratory System and the Cardiovascular System*[13]

Gambar 2 Sadapan Prekordial

1. Sadapan V1 ditempatkan di ruang intercostal IV di kanan sternum.
2. Sadapan V2 ditempatkan di ruang intercostal IV di kiri sternum.

3. Sadapan V3 ditempatkan di antara sadapan V2 dan V4.
4. Sadapan V4 ditempatkan di ruang intercostal V di linea (sekali pun detak apeks berpindah).
5. Sadapan V5 ditempatkan secara mendatar dengan V4 di linea axillaris anterior.
6. Sadapan V6 ditempatkan secara mendatar dengan V4 dan V5 di linea midaxillaris.

B. Sadapan Unipolar



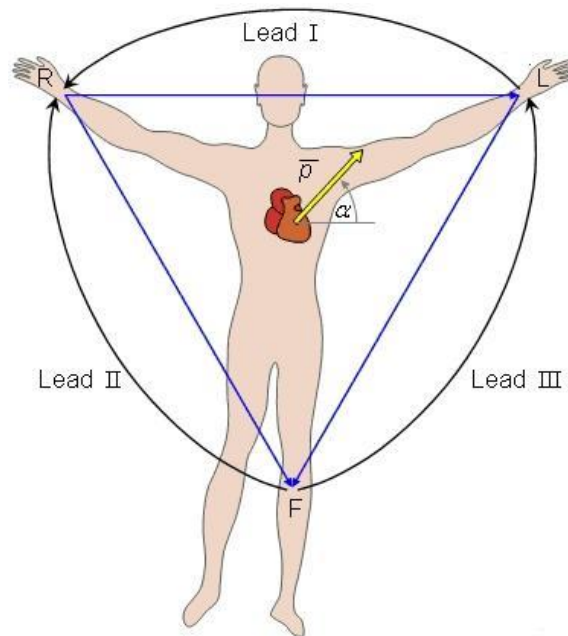
Sumber gambar : *The Respiratory System and the Cardiovascular System*[13]

Gambar 3 Sadapan Unipolar

1. aVR : merekam potensial listrik pada tangan kanan (RA) yang bermuatan (+), dan elektroda (-) gabungan tangan kiri dan kaki kiri membentuk elektroda indifiren.
2. aVL : merekam potensial listrik pada tangan kiri (LA) yang bermuatan (+), dan muatan (-) gabungan tangan kanan dan kaki kiri membentuk elektroda indifiren.
3. aVF : merekam potensial listrik pada kaki kiri (LF) yang bermuatan (+) dan elektroda (-) dari gabungan tangan kanan dan kaki kiri membentuk elektroda indifiren.

C. Sadapan Bipolar (Einthoven)

Merekam perbedaan potensial dari 2 elektroda, yang ditandai dengan angka romawi I, II dan III.



Sumber gambar : The Respiratory System and the Cardiovascular System

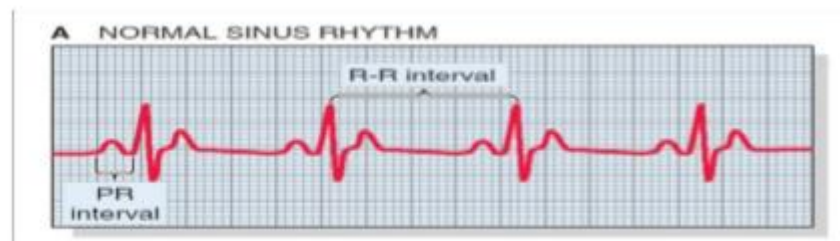
Gambar 4 Sadapan Bipolar

1. Sadapan I : merekam beda potensial antara tangan kanan (RA) yang bermuatan negatif (-) tangan kiri bermuatan positif (+).
2. Sadapan II : merekam beda potensial antara tangan kanan (-) dengan kaki kiri (LF) yang bermuatan (+).
3. Sadapan III : merekam beda potensial antara tangan kiri (LA) yang bermuatan (-) dan kaki kiri (+).

1.5. Irama Jantung

Irama jantung terdiri dari 3 macam yaitu Irama Sinus, Irama Junction, dan Irama Ventrikel. Masing-masing irama dinamai sesuai dengan asal impuls listrik yang keluar. Bila pencetus impuls listrik keluar dari SA Node maka irama yang muncul disebut Irama Sinus, dari SA Node muncul Irama Junction dan dari Ventrikel disebut Irama Idioventrikuler [1]. Untuk irama sinus yaitu irama jantung yang sumber pacemakernya berasal dari SA Node atau impulsnya berasal dari SANode, ada beberapa irama yang impulsnya berasal dari SA Node diantaranya :

1. Sinus Normal



Sumber gambar : Satu-satunya Buku EKG Yang Anda Perlukan (Edisi 7) [1]

Gambar 5 Ritme Sinus Normal

Ciri-ciri irama jantung normal :

- a. Gelombang P(+) membentuk gambar cembung seperti bukit.
- b. Kompleks QRS sempit tidak lebih dari 3 kotak.

2. Sinus Bradikardia



Sumber gambar : Satu-satunya Buku EKG Yang Anda Perlukan (Edisi 7) [1]

Gambar 6 Ritme Sinus Bradikardia

Ciri-ciri irama jantung bradikardia

- a. Irama teratur.
- b. RR interval jaraknya sama dengan 1 lead panjang.
- c. PP interval jaraknya sama dengan 1 lead panjang.
- d. Kompleks QRS harus sama dalam 1 lead panjang.
- e. Impuls dari SA node yang ditandai dengan adanya gelombang P yang mempunyai bentuk sama dalam 1 lead panjang.
- f. Frekuensi (HR) dibawah 60x/menit.
- g. Adanya gelombang P yang selalu diikuti kompleks QRS.
- h. Gelombang P dan kompleks QRS normal dan sama bentuknya dalam 1 lead.

Sinyal yang membedakan antara sinus bradikardi dengan normal sinus rhytm adalah frekuensi jantungnya. Pada sinus bradikardia frekuensi jantungnya kurang dari 60x/menit, sedangkan ciri yang lainnya persis sama dengan normal sinus rhytm.

Menemukan pasien dengan frekuensi jantung yang kurang dari 60x/menit dan bahkan kurang dari 50x/menit tanpa ada keluhan yang berarti, ini biasanya kita temukan pada jantung olahragawan atau orang yang melakukan kegiatan olah raga yang teratur seperti para atlit. Jadi siapapun orangnya yang melakukan olah raga secara teratur dan benar, akan didapatkan frekuensi jantung yang kurang dari 60 atau bahkan kurang dari 50 x/menit tanpa adanya keluhan. Frekuensi jantung yang kurang dari 60 x/menit juga bisa disebabkan oleh beberapa penyebab seperti obat-obatan (digoxin, beta blockers, calcium channel blockers) dan biasanya terjadi setelah pasca serangan jantung dan stroke. Jadi kita hanya melakukan intervensi pada pasien dengan frekuensi jantung yang kurang dari 60 x/menit apabila si pasien mengalami gejala-gejala yang mengganggu haemodinaik pasien. Untuk itu kita tidak boleh memutuskan hasil analisa ECG hanya dengan melihat ECG saja tanpa mempertimbangkan keadaan klinis pasien maupun riwayat kesehatan pasien.

3. Sinus Takikardia



Sumber gambar : Satu-satunya Buku EKG Yang Anda Perlukan (Edisi 7) [1]

Gambar 7 Ritme Sinus Takikardia

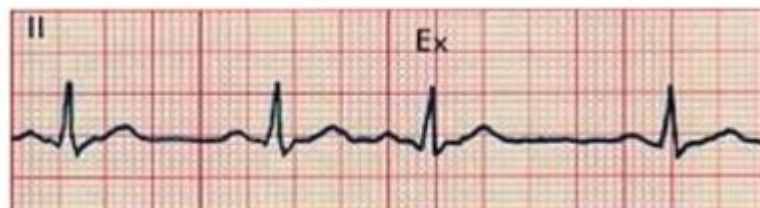
Ciri-ciri irama jantung takikardia :

Sama dengan sinus bradikardia, yang membedakanya adalah frekuensi jantung (HR) lebih dari 100x/menit. Seperti halnya dengan sinus bradikardia, hanya satu yang membedakan antara sinus takikardia dengan normal sinus rhythm yaitu frekuensi jantungnya. Pada sinus takikardia, frekuensi jantungnya lebih dari 100x/menit. Peningkatan frekuensi jantung pada pasien dengan sinus takikardia biasanya terjadi oleh adanya respon fisiologi dari tubuh terhadap suatu keadaan yang mengganggu hemostasis tubuh.

Dari sekian banyak penyebab frekuensi jantung melebihi 100x/menit pada kasus irama jantung takikardia, kita mengenal beberapa keadaan yang harus kita ketahui seperti:

- a. Sehabis melakukan aktivitas, demam, kesakitan, stress, dan cemas.
- b. Hypovolemik/terganggunya sistem sirkulasi darah ke jantung.
- c. Anemia.
- d. Obat-obatan.
- e. Gagal jantung.
- f. Pendarahan.

4. Sinus Aritmia



Sumber gambar : Satu-satunya Buku EKG Yang Anda Perlukan (Edisi 7) [1]

Gambar 8 Ritme Sinus Aritmia

Ciri irama jantung Aritmia :

Untuk membedakan sinus aritmia dengan normal sinus rhytm dan sinus bradikardia adalah pada sinus aritmia iramanya tidak teratur yang disebabkan oleh pengaruh pernapasan, baik inspirasi maupun ekspirasi. Biasanya kita akan menemukan pada anak ABG atau anak-anak yang masih muda.

1.6. Sensor EKG

Elektrokardiografi (EKG) merupakan suatu sinyal yang dihasilkan oleh aktivitas listrik otot jantung. EKG ini merupakan rekaman informasi kondisi jantung yang diambil dengan memasang elektroda pada tubuh. Rekaman EKG ini digunakan oleh dokter atau ahli medis untuk menentukan kondisi jantung dari pasien, yakni untuk mengetahui hal-hal seperti frekuensi (rate) jantung, arhythmia, infark miokard, pembesaran atrium, hipertrofi ventrikular, dll. Pemeriksaan EKG dilakukan dengan menempelkan elektroda (alat penerima impuls listrik jantung) di beberapa lokasi yang telah ditentukan. Setelah itu, informasi mengenai keadaan jantung dapat diketahui melalui pola grafik yang dihasilkan[13] .

Fungsi dasar dari elektroda adalah mendeteksi sinyal kelistrikan jantung. Dengan kata lain sebagai transduser untuk mengkonversi informasi biologis menjadi sinyal elektrik yang dapat diukur. Transduser ini dipakai dengan menggunakan interface jelly elektroda-electrolyte. Dengan menggunakan elektroda Ag/AgCl mengurangi noise dengan frekuensi rendah pada sinyal ECG yang terjadi karena pergerakan.



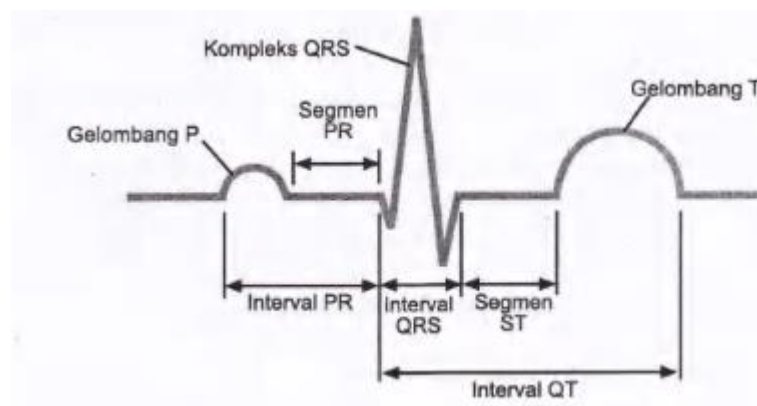
Gambar 9 Sensor EKG

1.7. Pengertian Metode QRS

QRS adalah metode yang digunakan untuk memproses sinyal yang dihasilkan pada saat melakukan tes EKG pada seseorang. Pada proses analisis metode QRS terbagi dalam 3 tahap diantaranya :

a) Signal Processing

Pada tahap ini dilakukan untuk mendeteksi adanya gangguan pada grafik EKG atau tidak. Setelah itu dilakukan proses analisis terhadap gelombang QRS dan dilakukan perbandingan pada setiap hasil yang keluar dari setiap lead untuk dilihat apakah ada hal yang tidak normal pada hasil test atau tidak.



Gambar 10 Grafik gelombang EKG

Gelombang P menggambarkan depolarisasi atrium. Gelombang ini kecil dan biasanya positif pada sadapan lateral kiri dan inferior. Gelombang ini sering bifasik pada sadapan III dan V₄. Gelombang ini selalu terlihat paling positif pada sadapan

II dan paling negatif pada sadapan AVR. Kompleks QRS menggambarkan depolarisasi ventrikel. Kompleks ini biasanya selalu positif pada sebagian besar sadapan lateral dan inferior. Di sepanjang daerah prekordium dari V' sampai Vu, ukuran gelombang R bertambah. Gelombang Q inisial kecil, yang menggambarkan depolarisasi septum, sering dapat ditemukan pada sadapan lateral kiri dan inferior. Gelombang I menggambarkan repolarisasi ventrikel. Gelombang ini merupakan bentuk gelombang yang paling bervariasi, tetapi biasanya positif pada sadapan dengan gelombang R tinggi.

Interval PR menggambarkan waktu mulai dari awal depolarisasi atrium sampai awal depolarisasi ventrikel. Segmen PR adalah waktu mulai dari akhir depolarisasi atrium sampai awal depolarisasi ventrikel. Interaal QRS menggambarkan durasi kompleks QRS. Segmen ST menggambarkan waktu mulai dari akhir depolarisasi ventrikel sampai awal repolarisasi ventrikel. Interaal QT menggambarkan waktu mulai dari awal depolarisasi ventrikel sampai akhir repolarisasi ventrikel. [1]

a) Rhythm Analysis

Pada tahap ini dilakukan analisis lebih lanjut ketika pada *signal processing* ditemukan hal abnormal pada gelombang EKG yang dihasilkan dengan cara di bandingkan dengan kriteria gelombang setiap penyakit yang ada.[6]

b) Diagnostic Interpretation

Pada tahap ini diagnosis dilakukan dengan tambahan variabel seperti umur, jenis kelamin, ras, klasifikasi klinis dan terapi obat dalam logikanya. Pada akhirnya pendekatan ini merupakan penyempurnaan dari diagnosis awal pada tahap *Rhythm Analysis*.

1.8. Pemograman Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar

kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dihubungkan) menjadi suatu kesatuan. Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu. [10]

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain. Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (Class)

Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara teknis kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, suatu organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan. Secara teknis, sebuah kelas saat program dieksekusi akan dibuat sebuah objek. Objek dilihat dari segi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi

sehingga sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Contoh metode atau operasi misalnya Read, Write, Move, Copy, dan sebagainya.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. *Reusability*

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

Komunikasi antarobjek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

1.9. Unified Modeling Language (UML)

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang. Banyak orang yang membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan

oleh banyak pihak adalah Data Flow Diagram (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman prosedural atau struktural. Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu Unified Modeling Language (UML) [11]. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut :

a. Use Case Diagram

Use Case atau diagram *Use Case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang akan ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *Use Case* yaitu pendefinisian apa yang disebut aktor dan *Use Case* [11].

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use Case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor.

b. Activity Diagram

Diagram aktivitas atau *Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut [11]:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem/user interface di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

c. *Class Diagram*

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

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

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsifungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebagai jenis-jenis kelas berikut [11]:

1. Kelas main Kelas yang memiliki fungsi awal dieksekusi ketika sistem berjalan.
2. Kelas yang menangani tampilan sistem Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *Use Case* Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *Use Case*.
4. Kelas yang diambil dari pendefinisian data Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

d. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Banyak diagram sekuen harus digambar adalah sebanyak pendefinisian *Use Case* yang memiliki proses sendiri atau yang penting semua *Use Case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *Use Case* yang akan didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak [11].

1.10. DBMS

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai Sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data [10].

Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

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

1.11. Java

Java adalah suatu teknologi di dunia software komputer, yang merupakan suatu bahasa pemrograman, dan sekaligus suatu platform. Sebagai bahasa pemrograman, Java dikenal sebagai bahasa pemrograman tingkat tinggi. Java mudah dipelajari, terutama bagi programmer yang telah mengenal C/C++. Java merupakan bahasa pemrograman berorientasi objek yang merupakan paradigma pemrograman masa depan. Sebagai bahasa pemrograman Java dirancang menjadi handal dan aman. Java juga dirancang agar dapat dijalankan di semua platform. Dan juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database. Sedangkan Java bersifat *neutralarchitecture*, karena Java Compiler yang digunakan untuk mengkompilasi kode program Java dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai Java Bytecode.

1.12. *Java Script Object Notation (JSON)*

Java Script Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, Java Script,

Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data. JSON terbuat dari dua struktur.

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash tabel*), daftar berkunci (*keyed list*) atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*) atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini. JSON menggunakan bentuk sebagai berikut.

1. Objek

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

2. Larik

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

3. Nilai

Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.

4. *String*

String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.

5. Angka

Fungsi Angka pada JSON sangat mirip dengan fungsi angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.

1.13. RethinkDB

RethinkDB adalah open source database pertama yang dibuat untuk aplikasi *realtime*. RethinkDB memungkinkan developer untuk membangun aplikasi *realtime* dalam waktu yang singkat dan usaha yang sedikit. Keuntungan dari RethinkDB adalah:

1. *Opensource Database* untuk membangun aplikasi web secara *realtime*.
2. *NoSQL Database* untuk menyimpan dokumen *schemaless JSON*.
3. Database yang terdistribusi dengan mudah untuk diukur [12]

1.14. Web Server

Tahun 1989, Tim Berners-Lee mengajukan pada perusahaannya, CERN (European Organization for Nuclear Research) sebuah proyek yang bertujuan untuk mempermudah pertukaran informasi antar para peneliti dengan menggunakan sistem hiperteks. Sebagai hasil atas implementasi proyek ini, tahun 1990 BernersLee menulis dua program komputer yaitu :

- a. sebuah peramban yang dinamainya sebagai WorldWideWeb.
- b. server web pertama di dunia, yang kemudian dikenal sebagai CERN httpd, yang berjalan pada sistem operasi NeXTSTEP.

Dari tahun 1991 hingga 1994, kesederhanaan serta efektifitas atas teknologi yang digunakan untuk berkunjung serta bertukar data melalui World Wide Web membuat kedua aplikasi tersebut diadopsi pada sejumlah sistem operasi agar dapat digunakan oleh lebih banyak individu, ataupun kelompok. Awalnya adalah organisasi penelitian, kemudian berkembang dan digunakan di lingkungan pendidikan tinggi, dan akhirnya digunakan dalam industri bisnis. Tahun 1994, Tim Berners-Lee memutuskan untuk membakukan organisasi World Wide Web Consortium (W3C) untuk mengatur pengembangan-pengembangan lanjut atas teknologi-teknologi terkait lainnya (HTTP, HTML, dan lain-lain) melalui proses standardisasi.

1.15. Web Service

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, maupun bahasa compiler.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya. Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

