BAB II

TINJAUAN PUSTAKA

II. 1. Tinjauan Tempat Penelitian

II. 1.1. Sejarah SMAN 6 Bandung

SMA Negeri 6 Bandung yang sekarang bertempat di Jalan Pasirkaliki No.51 Bandung berdiri sejak tahun 1956 dengan nama SMA Negeri “C” Bandung lokasi sekolah ini di Jalan Belitung Nomor 22 Berdasarkan SURAT KEPUTUSAN MENTERI P dan K tanggal 26 Oktober 1956 Nomor: 1956/E/III, di bawah pimpinan Bapak M.Sibaran.

Tahun 1966 SMA Negeri 6 Bandung kemudian berlokasi di Jalan Pasirkaliki 51, menempati gedung atau bangunan bekas sekolah Cina yaitu NAN HUA dengan nama SMA Negeri HOS COKROAMINOTO 51/400 Bandung. Ijin penggunaan bangunan sekolah Cina NAN HUA tersebut berdasarkan Surat Kepala Perwakilan Departemen P dan K Daerah Jabar tanggal 19 Desember 1966 Nomor 1866/H.2/S/66. SK ini juga berlaku bagi SMA Negeri 6 yang berlokasi di jalan Belitung 8/22 Bandung (SK. Pemecahan SMA Tahun Ajaran 1968dari Menteri Pendidikan dan Kebudayaan Republik Indonesia tanggal 20 November 1968 Nomor 374/U.K.K/3/1968).

Pada awal tahun 1976 SMA Negeri 6 hanya ada satu yaitu yang berlokasi di Jalan Pasirkaliki 51 Bandung (SK Kepala Kanwil Dept P dan K Propinsi Jawa Barat tanggal 10 Oktober 1975 No. 2553/A/1975) Kemudian SMA yang ada di Jalan Belitung menjadi SMA Negeri 9 Jalan Suparman Bandung.

Urutan kepemimpinan yang di mula tahun 1968 sampai tahun 2015 adalah sebagai berikut :

1. Bapak M.Sibarani, dari tahun 1956 sampai dengan Agustus 1968
2. Bapak Bukit Panggabean, Drs. Dari September 1968 sampai dengan Februari 1981
3. Bapak Sutarya Abdul Gani dari Maret 1981 sampai dengan Juli 1981 (Pejabat Sementara)
4. Bapak Drs, H. Sudiana AS,SH. Dari Agustus 1981 sampai dengan Juli 1986

5. Bapak Sap`an Sumarjaputra, Drs dari tanggal 20 Jul 1986 sampai dengan 30 Desember 1989
6. Bapak M. Soeparman, Drs. Mulai tanggal 2 Januar 1990 sampai dengan 2 Mei 1994
7. Bapak Maman, Drs (Kepala SMA Negeri 4) diangkat sebagai Pejabat Sementara mulai 3 Mei 1994 sampai dengan 31 Agustus 1994
8. Bapak Drs. M. Supomo, M.Pd. mulai dari tanggal 1 September 1994 sampai dengan 28 Februari 1996
9. Bapak Drs.R Y. Wardaya Kadarisman mulai 1 Maret 1996 sampai dengan 1 Nopember 1999
10. Bapak Drs. R. Kiryodono (Kepala SMU Negeri 4) diangkat sebagai Pejabat Sementara mulai 1 Nopember 1999 Sampai dengan 23 Maret 1999
11. Drs. H. Nana mulai 24 Maret 1999 sampai dengan 23 Maret 2002
12. Dra. Hj.Misbah Amin mulai 30 April 2002 sampai 30 Agustus 2003
13. Drs. Karyo Sunaryo mulai 1 April 2004 sampai dengan 30 Maret 2005
14. Drs. H. Uan yuhana mulai 1 April 2005 sampai dengan 30 Maret 2008
15. H. Cucu Saputra,M.M.Pd mulai 1 April 2008 sampai 31 Maret 2010.
16. Drs. Akhmad Rubandi SP. M.M.Pd mulai 1 April 2010 sampai dengan Maret 2012.
17. Drs. Ade Suryaman, S.Pd. mulai maret 2012 sampai dengan sekarang.

SMA Negeri 6 Bandung berlokasi di Jalan Pasirkaliki 51 Bandung, luas tanah 2.014 m2 dan merupakan tanah Eigendom dengan nomor Verponding 945, atas nama Elisabeth Clementine. SMA Negeri 6 Bandung berdiri dengan surat hak tanah tanggal 28 Mei 1934 No.444, Surat ukur tanggal 24 Desember No.174.

II. 1.2. Profil SMAN 6 Bandung

a. Visi

Terwujud sekolah menengah atas negeri 6 menjadi unggulan kota Bandung yang berwawasan lingkungan dengan mengembangkan potensi peserta didik sehingga berdaya saing, berprestasi, berkarya dan mampu menjawab tantangan perubahan zaman yang berdasarkan iman dan taqwa.

b. Misi

Misi dari SMAN 6 Bandung adalah sebagai berikut :

1. Meningkatkan kualitas keimanan dan ketaqwaan sehingga tercipta warga sekolah yang shaleh dan lingkungan yang religius.
2. Membekali peserta didik dengan ilmu pengetahuan, teknologi, seni dan keterampilan untuk mampu berkembang pada pendidikan yang lebih tinggi sesuai tuntutan kehidupan.
3. Membina peserta didik untuk mengembangkan dirinya agar dapat berprestasi sesuai dengan potensi, minat dan bakat yang dimilikinya.
4. Menumbuh kembangkan peserta didik sehingga mampu mandiri, berdaya cipta, belajar sepanjang hayat, untuk beradaptasi mengikuti perkembangan.
5. Meningkatkan kualitas kedisiplinan dan ketertiban peserta didik.
6. Meningkatkan kualitas daya saing peserta didik.
7. Meningkatkan profesionalisme pendidik dan tenaga ke pendidikan yang unggul dan bermutu.
8. Melengkapi sarana/prasarana sekolah sebagai daya dukung peningkatan mutu pendidikan.
9. Meningkatkan kesadaran di lingkungan sekolah akan pentingnya kebersamaan, kesehatan dan keindahan.
10. Membudayakan sekolah bersih, sehat dan indah sehingga mampu
11. meningkatkan kualitas belajar peserta didik.

c. Tujuan Pendidikan

Tujuan pendidikan di SMA Negeri 6 Bandung pada dasarnya mengacu pada tujuan pendidikan nasional dan visi, misi yaitu membangun manusia Indonesia seutuhnya sesuai dengan fitrahnya yaitu pribadi yang beriman dan bertaqwa terhadap Tuhan Yang Maha Esa, bermoral, berbudi pekerti luhur, berahlak mulia, demokratis menjunjung hak asasi manusia, menguasai ilmu pengetahuan, teknologi dan seni, memiliki kepribadian yang mantap dan mandiri serta memiliki tanggung jawab kemasyarakatan dan kebangsaan agar mampu mewujudkan kehidupan bangsa.

Tujuan pendidikan SMA Negeri 6 Bandung adalah :

1. Membekali warga sekolah yang memiliki kecerdasan emosional dan spiritual berdasarkan keimanan dan ketaqwaan kepada Tuhan Yang Maha Esa;
2. Menghasilkan lulusan yang memiliki kualitas kecerdasan intelektual yang tinggi, .sehingga memiliki daya saing secara kompetitif dan komparatif dalam persaingan tingkat lokal, regional maupun nasional.

II. 2. Landasan Teori

II. 2.1. Kualitas Perangkat Lunak

Definisi kualitas perangkat lunak menurut IEEE adalah sebagai berikut :

1. Sejauh mana sistem, komponen, atau proses dapat memenuhi kebutuhan yang telah ditentukan.
2. Tingkat di mana sebuah sistem, komponen, atau proses dapat memenuhi kebutuhan dan harapan dari pengguna.

II. 2. 2. 1. Model Kualitas ISO-9126

Model kualitas adalah landasan dari sistem evaluasi kualitas produk. Model kualitas menentukan karakteristik kualitas akan diperhitungkan ketika mengevaluasi sifat dari produk perangkat lunak.

Kualitas sistem adalah sejauh mana sistem memenuhi kebutuhan yang dinyatakan dan tersirat dari berbagai pemangku kepentingan, dan dengan demikian memberikan nilai. Kebutuhan tersebut *stakeholder* (fungsi, kinerja, keamanan, pemeliharaan, dll) yang tepat apa yang direpresentasikan dalam model kualitas, yang mengkategorikan kualitas produk menjadi karakteristik dan sub-karakteristik.

Tabel II-1 Faktor Dan Sub-Faktor Kualitas Internal dan Eksternal ISO-9126

Faktor	Sub-faktor
Functional	1. *Appropriateness* 2. *Accuracy* 3. *Interoperability* 4. *Security* 5. *Functional Compliance*
Reliability	1. *Maturity* 2. *Fault Tolerance* 3. *Recoverability* 4. *Reliability Compliance*
Efficiency	1. *Time-behaviour* 2. *Resource-utilisation* 3. *Efficiency Compliance*
Usability	1. *Understandability*

Faktor	Sub-faktor
	2. *Learnability* 3. *Operability* 4. *Attractiveness* 5. *Usability Compliance*
Maintainability	1. *Analyzability* 2. *Changeability* 3. *Stability* 4. *Testability* 5. *Maintability Compliance*
Portability	1. *Adaptabilitas* 2. *Installability* 3. *Co-existence* 4. *Replaceability* 5. *Transferability Compliance*

Berikut adalah pengertian dari masing-masing faktor dan sub-faktor dari Tabel II-1 Faktor Dan Sub-Faktor Kualitas Internal *dan Eksternal ISO-9126*, di antaranya sebagai berikut:

1. *Functional Suitability*

Functinal adalah tingkat di mana perangkat lunak dapat meniadakan fungsionalitas yang memenuhi kebutuhan ketika perangkat lunak digunakan pada kondisi tertentu.

a. *Suitability*

Suitability adalah tingkat dimana kemampuan dan kelayakan sebuah perangkat lunak untuk dapat menyediakan fungsionalitas untuk kebutuhan yang spesifik.

b. *Accucary*

Accuracy adalah tingkat dimana kemampuan sebuah perangkat lunak dapat memberikan hasilyang benar dan sesuai dengan fungsi yang dijalankan.

c. *Interoperability*

Interoperability tingkat dimana perangkat lunak memiliki kemampuan untuk berinteraksi dengan sistem yang lain atau sistem tertentu.

d. *Security*

Security merupakan sub-faktor yaitu merupakan tingkat keamanan yang dimiliki oleh perangkat lunak. Keamanan yang dimaksud dapat berupa pemberian hak akses kepada penggunanya.

e. *Functionality Compliance*

Functionality compliace menggambarkan tentagn kesesuaian antara fungsionalitas yang ada pada sebuah perangkat lunak dengan aturan standar yang ebrlaku pada perangkat lunak lainnya yang sejenis.

2. *Reliability*

Reliability merupakan tingkat sejauh mana produk perangkat lunak dapat mempertahankan tingkat tertentu ketika digunakan oleh pengguna dalam kondisi tertentu.

a. *Maturity*

Maturity merupakan sub-faktor yang berhubungan dengan tingkat dimana kelayakan sebuah perangkat lunak dalam menangani kegagalan atau kesalahan yang terdapat didalamnya.

b. *Fault Tolerance*

Fault Tolerance merupakan sub-faktor dari *realibility* dimana tingkat sebuah perangkat lunak dapat memberikan toleransi kegagalan atas kesalahan yang terjadi saat digunakan pada kondisi tertentu.

c. *Recoverability*

Recoveravility adalah tingkat dimana kemampuan sebuah perangkat lunak untuk dapat membangun kembali kinerja dan memulihkan data baik secara langsung maupun secara tidak langsung.

d. *Reliability Compliance*

Reliability compliance merupakan tingkat di mana perangkat lunak dapat memenuhi standar ketahanan yang dimiliki oleh perangkat lunak lain sejenis.

3. *Efficiency*

Performance efficiency merupakan tingkatan sejauh mana produk perangkat lunak memberikan kinerja yang tepat, dengan jumlah sumber daya yang digunakan, dalam kondisi tertentu.

a. *Time-behaviour*

Time-behaviour yaitu waktu yang dibutuhkan oleh sebuah perangkat lunak untuk menjalankan suatu fungsi atau tujuan tertentu.

b. *Resource-utilisation*

Resource-utilisation merupakan tingkat dimana jumlah sumber daya yang digunakan oleh perangkat lunak untuk melakukan fungsi atau tujuan tertentu.

c. *Efficiency Compliance*

Efficiency compliance merupakan tingkatan di mana perangkat lunak memenuhi standar yang berhubungan dengan efisiensi kinerja perangkat lunak.

4. *Usability*

Usability merupakan tingkat di mana perangkat lunak dapat dipahami, dipelajari, digunakan dan menarik untuk pengguna, jika digunakan dalam kondisi tertentu.

a. *Understandability*

Undersatandability merupakan sub-faktor yang menggambarkan tingkat dimana seberapa jauh perangkat lunak dapat dipahami oleh pengguna baik dari secara konsep logis dan penerapan penggunaan perangkat lunak tersebut.

b. *Learnability*

Learnability merupakan tingkat dimana sebuah perangkat lunak daat dipelajari dengan baik oleh penggunanya.

c. *Operability*

Operability yaitu tingkat dimana seberapa jauh perangkat lunak dapat dioperasikan oleh penggunanya.

d. *Attractiveness*

Attractiveness merupakan tingkatan di mana perangkat lunak dapat menarik perhatian bagi para pengguna.

e. *Usability Compliance*

Operability compliance merupakan tingkat di mana perangkat lunak dapat memenuhi standarisasi dan panduan yang berhubungan dengan penggunaan perangkat lunak.

5. *Maintability*

Maintability merupakan tingkatan sejauh mana produk perangkat lunak dapat dimodifikasi. modifikasi mungkin termasuk koreksi, perbaikan atau adaptasi dari perangkat lunak untuk perubahan lingkungan, dan persyaratan dan spesifikasi fungsional.

a. *Analyzability*

Analyzability merupakan tingkat mana perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak atau untuk mengidentifikasi bagian yang dapat dimodifikasi.

b. *Changeability*

Changeability merupakan tingkat di mana perangkat lunak memungkinkan sebuah modifikasi yang spesifik untuk dapat diimplementasikan. Hal ini memudahkan perangkat lunak dapat dimodifikasi.

c. *Stability*

Modification stability merupakan tingkat di mana perangkat lunak untuk dapat menghindari efek yang tidak diharapkan dari modifikasi yang dilakukan terhadap perangkat lunak.

d. *Testability*

Testability merupakan tingkat di mana perangkat lunak memungkinkan modifikasi perangkat lunak untuk dilakukan validasi.

e. *Maintainability Compliance*

Maintability compliance merupakan tingkat di mana perangkat lunak dapat memenuhi standarisasi yang berhubungan dengan *maintainability* pada perangkat lunak lain.

6. *Portability*

Transferability merupakan tingkatan sejauh mana produk perangkat lunak dapat disesuaikan ketika perangkat lunak berpindah dari lingkungan yang satu ke lingkungan yang lain.

a. *Adaptability*

Adaptability merupakan tingkat di mana perangkat lunak dapat beradaptasi pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

b. *Installability*

Installability merupakan tingkat di mana perangkat lunak dapat dipasang dan dihapus pada lingkungan yang spesifik.

c. *Co-extistence*

Co-existence merupakan tingkat di mana perangkat lunak dapat berdampingan dengan perangkat lunak lain dan berbagi sumber daya pada lingkungan yang sama tanpa ada dampak yang merugikan [4].

d. *Replaceability*

Replaceability merupakan tingkat di mana perangkat lunak dapat digunakan pada perangkat lain dengan tujuan yang spesifik pada lingkungan yang sama [4].

e. *Portability Compliance*

Transferability compliance merupakan tingkat di mana sebuah perangkat lunak dapat memenuhi standar yang berhubungan dengan kemampuan perangkat lunak untuk dapat berpindah dari satu lingkungan ke lingkungan lain.

II. 2. 2. 2. Penilaian Standar Kode Program

Penilaian standar kode program adalah menguji validitas dari kode program satu *website*. Pengujian dilakukan dengan cara melihat kode program yang ada dengan standar *website*, standar *website* menurut Eric Meyer adalah sebuah maksud agar satu *website* memiliki kode program sesuai standar yang ada dari beberapa organisasi seperti w3, alasannya adalah agar *browser* dan perangkat lunak lainnya dapat memahami kode program yang ada pada *website* dengan lebih cepat.

Standar *website* adalah alat yang digunakan untuk menyetujui dan menerima *website* sebagai alat komunikasi yang luas dan dapat diakses oleh berbagai pengguna dan berbagai perangkat. Karakteristik umum untuk standar *website* adalah sebagai berikut :

a. *Semantically Correct Markup*

Semantically correct markup yang benar adalah menggunakan elemen HTML untuk tujuan yang sesuai artinya ketika menulis kode program HTML, elemen-elemen HTML harus digunakan contohnya seperti penggunaan elemen H1 untuk judul, menggunakan elemen paragraf untuk teks paragraf, menggunakan elemen *list* untuk daftar.

b. *Valid Code*

Valid code adalah adalah penggunaan kode program yang valid sesuai dengan standar yang formal, seperti yang diterbitkan oleh W3C. Alasan kenapa harus

menggunakan kode program yang valid salah satu alasannya adalah kode program yang valid akan membuat *website* lebih cepat daripada kode program yang memiliki kesalahan.

c. *Accessible Code (For Humans And Devices)*

Maksud *Accessible Code (For Humans And Devices)* adalah kode pemrograman yang mudah diakses (untuk manusia dan perangkat komunikasi).

d. *CSS To Separate Content From Presentation*

Cascading Style Sheets (CSS) To Separate Content From Presentation adalah memisahkan antara berkas CSS dari berkas HTML.

Penilaian kode program akan dilakukan dengan menggunakan alat pengukuran yang bertujuan mendapatkan pengecekan *eror*, validitas sumber kode program, kecepatan yang dimiliki oleh *website.* Alat pengukuran terdiri dari dua jenis yaitu alat pengukuran *online* dan alat pengukuran *offline*. Alat pengukuran Online yang digunakan adalah *website* http://validator.w3.org/ untuk menguji validitas dari kode HTML dan http://jigsaw.w3.org/css-validator/ untuk menguji validitas kode CSS. Alat pengukuran *offline* adalah dengan cara melakukan observasi secara langsung kepada sumber kode program dan akan dilakukan perbandingan dengan standar kode program. Kode program yang dibandingkan secara manual adalah kode program PHP .

Standar kode program yang digunakan untuk kode program PHP adalah menggunakan standar dari *framework* PHP yaitu *CodeIgniter*.

II. 2. 2. 3. Kesalahan Kode HTML

Kesalahan kode html merupakan kesalahan pada saat penulisan kode dan penggunaan elemen-elemen html yang tidak sesuai, berikut adalah kesalaha-kesalahan yang dapat terjadi pada saat penulisan kode html :

1. *End Tag For X Omitted, But OMITTAG NO Was Specified*

Kesalahan jenis ini terjadi karena tidak ditutup dengan benar sebuah *tag* elemen di mana sebuah *tag* elemen yang ditulis dengan kondis*i self-close* tetapi tidak ditutup oleh “/>” tetapi menggunakan “>”.

2. *Reference To Entity X For Which No System Identifier Could Be Generated*

Kesalahan pada jenis ini karena penggunaan sebuah *identifier* yang tidak dapat dihasilkan sehingga tidak terjadi perintah apapun.

3. *Document Type Does Not Allow Element X Here; Missing One Of Y Start-Tag*

Kesalahan jenis ini adalah karena penggunaan elemen *block-level* yang tidak sesuai, dan juga kesalahan karena tidak menutup elemen tersebut. Contoh seperti elemen *block – level* <*p*> atau <*table*> yang berada di dalam elemen *inline* seperti <*span*> atau <*font*>.

4. *An Attribute Value Must Be A Literal Unless It Contains Only Name Characters*

Jenis kesalahan ini terjadi karena penggunaan karakter yang tidak tepat khususnya pada penggunaan "*name characters*" sehingga tidak dapat dikenali dalam penggunaan nilai atribut di mana sebuah nilai atribut yang mengandung huruf a-z harus menggunakan tanda kutip. Tidak hanya untuk yang hanya mengandung huruf tetapi yang tidak mengandung huruf lebih baik menggunakan tanda kutip.

5. *End Tag For X Which Is Not Finished*

Jenis kesalahan ini karena penulisan kode pada tak yang bersarang ternyata *tag child* belum ditutup dan ternyata *tag parent* sudah ditutup. Contoh seperti elemen <*html*> didalam nya terdapat tag <*title*>, *tag title* harus ditutup lebih dulu </*title*> baru *tag* <*html*> dapat ditutup </*html*>.

6. *There Is No Attribute X*

Jenis kesalahan ini adalah penggunaan atribut yang tidak sesuai atau atribut yang sudah tidak didukung pada saat penggunaan atribut pada sebuah elemen di dokumen *html* tersebut.

7. *An Attribute Value Specification Must Be An Attribute Value Literal Unless SHORTTAG YES Is Specified*

W3.org belum mengkonfirmasi kesalahan ini secara resmi tetapi jika dilihat dari pesan kesalahan yang merujuk pada penulisan kode di mana nilai sebuah atribut haru mengandung nilai *literal* kecuali atribut tersebut memiliki nilai yang sudah spesifik.

8. *General Entity X Not Defined And No Default Entity*

Jenis Kesalahan ini adalah penggunaan "&" pada URL untuk mengacu pada sebuah entitas.

9. *Required Attribute X Not Specified*

Jenis Kesalahan berikut terjadi karena penulisan kode yang tidak sesuai dengan tidak menuliskan atribut yang seharusnya terdapat pada sebuah *tag html*, contohnya seperti *tag img* pada *html* di mana seharusnya terdapat atribut *alt* pada elemen tersebut.

10. *Document Type Does Not Allow Element X Here; Assuming Missing Y Start-Tag*

W3.org belum mengkonfirmasi secara resmi terkait kesalahan ini, tapi jika dilihat dari kesalahan yang terjadi dapat diambil kesimpulan bahwa kesalahan jenis ini terjadi karena penggunaan elemen yang tidak sesuai pada tempatnya karena elemen tersebut tidak tepat untuk digunakan pada dokumen tersebut dan memberikan asumsi dengan penggunaan yang lebih tepat.

11. *Element X Undefined*

Jenis kesalahan ini disebabkan oleh penggunaan atribut yang tidak sesuai dengan fungsinya, dan fungsi sudah mendapatkan pembaharuan sehingga fungsi yang lama sudah digantikan.

12. *Character X Not Allowed In Attribute Specification List*

Jenis kesalahan ini terjadi karena penggunaan sebuah karakter yang tidak tepat atau tidak diperbolehkan berada pada baris kode tersebut.

13. *ID X Already Defined*

Jenis kesalahan ini adalah karena kode html memiliki *id* yang tidak unik dan digunakan untuk berkaitan dengan *style sheets*, karena lebih tepat jika menggunakan *class* dari pada *id* untuk *style sheets* pada kode html.

14. *X Not Finished But Containing Element Ended*

Jenis kesalahan ini adalah karena terdapat sebuah *tag* atau elemen yang belum selesai atau belum di tutup tetapi sudah ditutup oleh elemen lain. Contoh seperti elemen <*ul*> belum ditutup oleh </*ul*> tetapi sudah ditutup lebih dulu oleh elemen </*div*>.

15. *Document Type Does Not Allow Element X Here*

Kesalahan jenis ini adalah karena penggunaan atribut yang tidak tepat atau tidak ditempatkan di tempat yang seharusnya.

II. 2.2. *Code Refactoring*

Metode refactoring adalah metode menulis ulang sebuah kode program dengan cara memperbaiki sumber kode program tanpa mengubah *behaviour* sumber kode program sebelumnya, hal ini ditujukan untuk meningkatkan struktur dari kode program itu sendiri agar lebih baik. Pada penelitian ini code refactoring digunakan untuk membuat website sesuai dengan standar. Website standar dimaksudkan untuk menjadi dasar umum pada sebuah website sehingga browser dan perangkat lunak lain dapat memahami dengan mudah sebuah kode program diterjemahkan.

Standar kode program yang digunakan pada penelitian ini adalah sebagai berikut :

a. CodeIgniter *Style Guide*

Standar kode program CodeIgniter ini digunakan untuk standar kode program *hypertext preprocessor* (PHP).

b. Google HTML/CSS *Style Guide*

Panduan standar kode pemograman Google digunakan untuk standar kode program *hypertext markup language* (HTML) dan *cascading style sheets* (CSS).

Menurut Martin Fowler langkah untuk melakukan refactoring adalah sebagai berikut :

1. Melakukan Tes Terhadap Kode

Tes yang dilakukan terhadap kode dilakukan untuk mencari tahu letak kesalahan sumber kode program. Tes harus dilakukan sebaik mungkin sehingga dapat mengetahui dengan baik kesalahan yang terdapat pada sumber kode program tersebut. Menguji validasi standar kode program bisa menjadi salah satu tes yang bisa digunakan untuk mengetahui kesalahan yang terdapat pada sumber kode program.

2. Melakukan Perbaikan Sumber Kode Program

Setelah menemukan kesalahan pada sumber kode program, sumber kode program tersebut diperbaiki. Perbaikan dilakukan hingga ketika dilakukan tes kembali terhadap sumber kode program tidak menemukan kesalahan pada sumber kode program.

II. 2.3. Metode *Rank Order Centroid*

Metode *rank order centroid* merupakan sebuah metode pembobotan yang terbilang sederhana. Maksud sederhana adalah cara yang digunakan pada metode ini untuk memberikan bobot terhadap sejumlah item menurut kepentingtan dari item-item tersebut. Pada penelitian ini pembobotan diberikan pada sub-faktor eksternal ISO-9126. Berikut adalah persamaan yang digunakan untuk memberikan bobot kepada setiap faktor dan sub-faktor tersebut.

$$W_i = \left(\frac{1}{M}\right) \sum_{n=i}^{M} \frac{1}{n} \qquad \text{(II-1)}$$

Dimana :

W_i = Bobot ke i

M = *Number of Item*

II. 2.4. Metode *Weighted Summation* (WSum)

Weigth Summation adalah sebuah metode kompensasi, yang berarti dapat menghitung skor kriteria skor yang "buruk", sehingga dapat dikopensisasikan dengan kriteria skor yang "baik". Berikut adalah persamaan yang digunakan dalam metode *weighted Summation.*

Jika

$$score\ (Ci) = \sum_{i=1}^{n} Wi.Vi\ (Ci) \qquad \text{(II-2)}$$

Maka

$$score\ (Ci) = \sum_{i=1}^{n} Wi.Xi\ (Ci) \qquad \text{(II-3)}$$

Dimana :

score(Ci) = Nilai kriteria Ci ($0 \leq$ score Ci ≥ 1)

Wi = Bobot untuk Kriteria Ci

Xi = Nilai Perhitungan metrik kriteria ke-i

N = Banyaknya Kriteria

II. 2.5. *Database*

Database merupakan sebuah kumpulan data-data yang dimiliki oleh suatu program yang disimpan pada sebuah *file*. *Database* terdiri dari tabel-tabel yang memiliki *field* di dalamnya yang menyimpan sebuah data. Seperti sebuah program kasir yang memiliki *database* penjualan di mana di dalamnya terdapat tabel-tabel yang menyimpan data transaksi dan data barang. Sebuah *database* pada umumnya disimpan pada sebuah media penyimpanan seperti *harddisk* tapi masa kini juga terdapat *database* yang disimpan pada media penyimpanan *cloud*. Sebuah *database* memiliki sebuah skema. Terdapat dua jenis *database* yaitu *database relasional* dan *database* tidak *relasional*/*nosql*. Janis dari *database* tersebut dilihat dari skema yang digunakan pada struktur tabel yang ada pada *database* tersebut. *Database relasional* adalah sebuah *database* di mana setiap tabel memiliki relasi dengan tabel lain.

II. 2. 4. 1. Normalisasi Database

Normalisasi *database* merupakan sebuah metode untuk membuat sebuah tabel pada struktur *database* menjadi normal dengan tujuan untuk mengoptimalkan *database* salah satu caranya dengan meminimalkan adanya data *redudansi*. Terdapat langkah-langkah ketika melakukan normalisasi, langkah-langkah tersebut adalah sebagai berikut :

1. Bentuk normal pertama

Pada bentuk normal pertama tabel-tabel pada *database* tersebut akan dilihat apakah memiliki kolom yang memiliki sifat yang sama atau tidak, jika jawabannya iya maka kolom tersebut dihilangkan hingga sisa satu kolom saja. Lalu setiap tabel dibuat berdasarkan setiap kelompok data yang terkait. Selanjutnya buat identitas untuk setiap data *record* yang biasa disebut sebagai *primary key*.

2. Bentuk normal kedua

Pada bentuk normal kedua ini setiap isi pada sebuah kolom berulang maka dibuat *subtabel* dari tabel tersebut, serta tabel tersebut harus dibuat sebuah relasi menggunakan *foreign key*.

3. Bentuk normal ketiga

Pada bentuk normal ketiga ini setiap tabel harus sudah memenuhi aturan dari bentuk normal kedua dan menghilangkan kolom yang tidak berhubungan dengan *primary key*.

II. 2.6. *Responsive website*

Design web responsif adalah sebuah pendekatan *design* sebuah web yang bertujuan untuk memberikan tampilan dan interaksi yang optimal. Interaksi optimal adalah di mana pengguna dapat mudah membaca isi web serta navigasi yang nyaman walaupun resolusi berubah karena diakses dari perangkat berbeda seperti desktop atau monitor komputer dan perangkat *mobile*. Desain responsif untuk membuat nya diperlukan tiga kondisi di antaranya :

1. *A Flexible, grid-based layout*
2. *Flexible image and media*
3. *Media queries*