

The background of the page is a repeating pattern of a watermark logo. The logo consists of a stylized gear or circular shape with three colored segments (blue, yellow, and grey) and the text 'POLITEKNIK NEGERI CILACAP' below it.

BAB II

TINJAUAN PUSTAKA

DAN LANDASAN TEORI

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai sistem penunjang keputusan menggunakan metode topsis untuk menentukan kelayakan bantuan rumah tidak layak huni pada desa Sumbaga dilakukan oleh Hiya Nalatissifa dan Yudi Ramdhani (2020). Permasalahan yang dialami yaitu pihak desa kesulitan dalam hal penetapan penerima bantuan dikarenakan banyaknya calon penerima dan kriteria yang harus dipertimbangkan dalam pengolahan datanya, keputusan yang diambil hanya mengandalkan prediksi/perkiraan sehingga masih terdapat bantuan yang tidak tepat sasaran. Tujuan penelitian ini membangun aplikasi sistem penunjang keputusan untuk mempermudah desa Sumbaga dalam proses seleksi kelayakan bantuan RTLH[2].

Penelitian serupa dilakukan oleh Yustika Indah Purwanti, dkk (2021). Pada sistem berjalan saat ini pihak Dinas Sosial masih mengalami kesulitan seperti dalam pengolahan datanya membutuhkan ketelitian, sehingga memungkinkan terjadinya rangkap data dan juga kesalahan dalam penentuan penduduk yang harus diutamakan, sehingga diperlukannya suatu sistem pendukung keputusan untuk menentukan siapa yang berhak untuk didahulukan dalam mendapatkan bantuan RTLH[3].

Penelitian sebelumnya dilakukan oleh Shinta Siti Sundari, dkk (2019). Proses pelaksanaan yang berjalan pada saat ini yaitu Dinas Sosial Kabupaten Tasikmalaya merekomendasikan penerima bantuan RTLH diawali dengan pengumpulan data calon penerima bantuan rehabilitasi RTLH yang diketahui dan direkomendasikan oleh Camat dan Kepala Desa setempat berdasarkan kriteria dari Dinas Sosial. Dengan diterapkannya sistem pendukung keputusan pemilihan penerima bantuan rehabilitasi rumah tidak layak huni berbasis web di Dinas Sosial Kabupaten Tasikmalaya diharapkan mampu meminimalisir permasalahan-permasalahan seperti yang sudah dipaparkan[4].

Pada penelitian yang dilakukan oleh Yulisman dan Febriani (2020). Proses penilaian penentuan penerima bantuan rumah sehat layak huni tersebut belum sesuai dengan kriteria dan hanya berdasarkan laporan Ketua Rukun Tetangga (RT) tanpa dilengkapi data pendukung yang valid. Sehingga peneliti membangun sistem pendukung keputusan untuk menentukan penerima bantuan rumah sehat layak huni sesuai dengan

kriteria dan tepat sasaran. Adapun kriteria yang digunakan antara lain, luas ruangan, kondisi dinding, kondisi atap, kondisi lantai, usia, kepemilikan tanah, tanggungan keluarga, penghasilan, lama berdomisili, status perkawinan, pekerjaan, kesehatan, kepemilikan rumah. Peneliti dalam membangun Sistem pendukung keputusan menggunakan metode SAW. Dengan adanya sistem ini, pada saat menentukan penerima bantuan dapat lebih cepat dibandingkan dengan sistem yang sedang berjalan saat ini[5].

Pada penelitian ini dilakukan pada Desa Cangkring yang dilakukan oleh Ahmad Jazuli dan Mukhamad Nurkamid (2019). Tujuan dibangunnya sistem pendukung keputusan di Desa Cangkring untuk menentukan pemberian bantuan kepada warga desa. Peneliti dalam membangun sistem ini menggunakan metode *Multifactor Evaluation Process* (MFEP) untuk menentukan bantuan pemugaran rumah tidak layak huni sesuai dengan kriteria kepemilikan rumah, status dinding rumah, status lantai rumah, status atap rumah, status luas bangunan dan status penghasilan[6].

Berdasarkan perbedaan dari kajian penelitian diatas, maka peneliti akan membuat Sistem Pendukung Keputusan Kelayakan Penerima Bantuan Rumah Tidak Layak Huni Di Kabupaten Cilacap Dengan Menggunakan Metode TOPSIS. Sistem pendukung keputusan ini dibangun menggunakan metode *user centered design*, native, menggunakan Bahasa pemrograman PHP, *database developer* MySQL, web server apache dan tools editor sublime text. Hasil dari penelitian ini adalah sebuah sistem yang dapat memudahkan kepala DISPERKIMTA dalam proses pengecekan data dan hasil akhir dari penentuan penerima bantuan RTLH, memudahkan staff DISPERKIMTA meminimalisir kesalahan dalam penentuan calon penerima bantuan, serta memudahkan staff kecamatan dalam mengumpulkan dan mengelola data calon penerima bantuan. Sehingga memudahkan tim penyeleksi dalam mengambil keputusan secara cepat dan tepat.

2.2 Landasan Teori

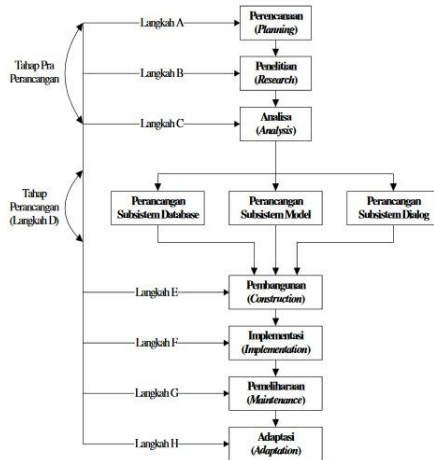
2.2.1 Sistem Pendukung Keputusan

Sistem pendukung keputusan merupakan sistem informasi komputer yang menghasilkan berbagai permasalahan semi terstruktur ataupun tidak terstruktur dengan menggunakan data dan model[7]. Aplikasi ini biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk satu peluang[8]. Sistem pendukung keputusan sebagai

sistem komputer yang terdiri dari tiga komponen yang saling berinteraksi, sistem bahasa (mekanisme untuk pendukung keputusan lain), sistem pengetahuan (repositori pengetahuan domain masalah yang ada pada sistem pendukung keputusan atau sebagai data) dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambil keputusan). Keputusan yang diberikan dapat meningkatkan efektivitas pengambilan keputusan dan pengambilan keputusan dapat lebih objektif[2].

Sistem pendukung keputusan memiliki sifat khas atau karakteristik dalam membantu mengambil keputusan yaitu:

1. Mendukung proses pengambilan keputusan suatu organisasi atau perusahaan
2. Adanya *interface* manusia atau mesin dimana manusia atau *user* tetap memegang kontrol.
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur serta mendukung beberapa keputusan yang saling berinteraksi.
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai kebutuhan.
5. Memiliki subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan sistem.
6. Memiliki dua komponen utama yaitu data dan model yang menggambarkan proses pengambilan keputusan dari empat fase. Untuk membangun sistem pendukung keputusan dikenal dengan delapan tahapan seperti pada gambar[9]. Delapan tahapan perancangan sistem pendukung keputusan antara lain:



Gambar 2. 1 Tahapan perancangan spk

1. Perencanaan (*Planning*)
Perencanaan pada umumnya berhubungan dengan perumusan masalah serta penentuan tujuan dari sistem pendukung keputusan.
2. Penelitian (*Research*)
Penelitian berhubungan dengan pencarian data serta sumber daya yang tersedia.
2. Analisa (*Analysis*)
Tahap ini termasuk penentuan teknik perancangan dan pendekatan pengembangan sistem yang akan dilakukan serta sumber data yang dibutuhkan.
3. Perancangan (*Design*)
Tahap ini dilakukan perancangan terhadap ketiga sub sistem dari sistem pendukung keputusan yaitu sub sistem database, sub sistem model, dan sub sistem dialog.
4. Pembangunan (*Construction*)
Pembangunan merupakan kelanjutan dari tahap perancangan, dimana ketiga sub sistem yang dirancang digabungkan menjadi

suatu sistem pendukung keputusan. Pada tahap ini dimulai penulisan Bahasa pemrograman bagi sistem pendukung keputusan.

5. Implementasi (*Implementation*)

Tahap ini merupakan penerapan sistem pendukung keputusan yang dibangun, yang terdapat beberapa tugas yang harus dilakukan seperti melakukan pengecekan, evaluasi, demonstrasi, orientasi, pengembangan.

6. Pemeliharaan (*Maintenance*)

Pada tahap ini melibatkan perencanaan dukungan yang harus dilakukan terus menerus untuk mempertahankan kendala sistem.

7. Adaptasi (*Adaptation*)

Hal ini dilakukan pengulangan terhadap tahap-tahap diatas sebagai tanggapan atas perubahan kebutuhan pengguna.

Metode yang digunakan dalam sistem pendukung keputusan ini adalah metode TOPSIS. Metode *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) merupakan sebuah metode yang mampu membantu proses pengambilan keputusan yang optimal untuk menyelesaikan masalah keputusan secara praktis[11]. Hal ini karena konsep yang mudah dipahami dan sederhana, komputasi yang efisien, dan mampu mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis sederhana[2]. Prinsip metode ini adalah alternatif yang dipilih memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif[12].

Penyelesaian metode TOPSIS sebagai berikut[13]:

1. Membuat matriks keputusan yang ternormalisasi.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$.

r_{ij} = matriks keputusan ternormalisasi

x_{ij} = bobot kriteria ke j pada alternatif ke i

i = alternatif ke i

j = alternatif ke j

2. Membuat matriks keputusan yang ternormalisasi terbobot.

$$y_{ij} = w_{ij}r_{ij}$$

Dengan $i = 1, 2, \dots, m$; dan $j = 1, 2, \dots, n$.

y_{ij} = matriks ternormalisasi bobot

w_{ij} = matriks keputusan ternormalisasi

r_{ij} = bobot kriteria ke- j

Dimana w_j adalah bobot dari kriteria ke- j .

3. Membuat matriks solusi ideal positif dan matriks solusi ideal negatif. Berdasarkan rating bobot ternormalisasi maka dapat menentukan solusi ideal positif (A^+) dan solusi ideal negatif (A^-) untuk dapat menentukan solusi ideal sebelumnya harus ditentukan apakah atribut bersifat keuntungan (*benefit*) atau bersifat biaya (*cost*).

$$A^+ = (y_{1+}, y_{2+}, \dots, y_{n+})$$

$$A^- = (y_{1-}, y_{2-}, \dots, y_{n-})$$

Dimana,

$$\left. \begin{aligned} y_j^+ &= \max_i y_{ij} \text{ jika } j \text{ adalah atribut keuntungan} \\ &= \min_i y_{ij} \text{ jika } j \text{ adalah atribut biaya} \end{aligned} \right\}$$

$$\left. \begin{aligned} y_j^- &= \min_i y_{ij} \text{ jika } j \text{ adalah atribut keuntungan} \\ &= \max_i y_{ij} \text{ jika } j \text{ adalah atribut biaya} \end{aligned} \right\}$$

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan negatif.

$$D_i^+ = \sqrt{\sum_j^n = 1 (y_j^+ - y_{ij})^2}$$

$$D_i^- = \sqrt{\sum_j^n = 1 (y_{ij} - y_j^-)^2}$$

Menentukan nilai prefensi untuk setiap alternatif.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+}$$

Keterangan:

D_i^+ adalah jarak antara nilai alternatif ke i dengan solusi ideal positif

D_i^- adalah jarak antara nilai alternatif ke i dengan solusi ideal negatif

2.2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak didefinisikan sebagai penerapan suatu pendekatan sistematis, disiplin dan terkuantifikasi atas pengembangan, penggunaan dan pemeliharaan perangkat lunak, serta studi atas pendekatan-pendekatan ini, yaitu penerapan pendekatan *engineering* atas perangkat lunak.

Perangkat lunak dikembangkan dengan rekayasa perangkat lunak yang didalamnya mencakup hal-hal berikut ini:

1. Proses, urutan langkah-langkah dalam penciptaan perangkat lunak akan melibatkan analisis dan pemodelan formal maupun informal pada proses-proses perangkat lunak.
2. Kebutuhan, teknik-teknik analisis dibutuhkan untuk keutuhan prioritas dan menentukan konflik kebutuhan.
3. Arsitektur, pemodelan sistem perangkat lunak pada tingkat tinggi dari abstraksi.
4. Desain, metodologi yang memungkinkan sebuah masalah besar yang terstruktur untuk diuraikan ke dalam modul-modul dengan menggunakan bahasa pemrograman.
5. Pengujian, teknik-teknik untuk melatih sistem perangkat lunak untuk menentukan terpenuhinya kebutuhan aplikasi.
6. Lingkungan, perangkat lunak yang digunakan oleh pengembang didalam konstruksi sistem perangkat lunak.

Dengan ini peneliti menggunakan metode pengembangan sistem *user centered design*. Adapun jalan penelitian menggunakan metode UCD yaitu:

1. *Plan the human centred process*

Tahapan pertama akan dilakukan analisis data maupun teori yang berkaitan dengan proses membangun sistem yang berpusat kepada pengguna. analisis data teori, metode yang memiliki tujuan untuk

menggal kebutuhan yang diperlukan untuk penelitian ini dan seluruh kebutuhan yang terkait dengan penelitian ini.

2. *Specify the context of use*

Dari tahap ini akan diidentifikasi siapa saaa user yang akan menggunakan produk. Kegiatan observasi dan wawancara dilakukan antara peneliti sebagai pengembang sistem terkait dengan sistem yang akan dibangun dan diimplementasikan nantinya sehingga mendapatkan gambaran awal mengenai website sesuai dengan kebutuhan *user*.

3. *Specify user and organizational requirements*

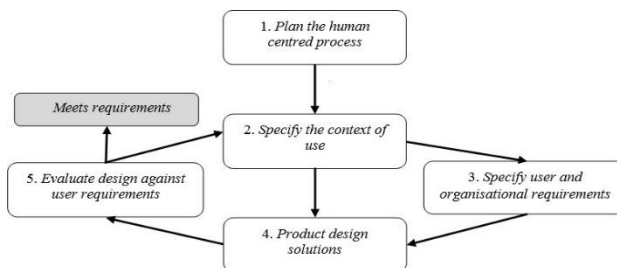
Kegiatan pada tahapan ini adalah penggalan informasi atau data untuk mengumpulkan kebutuhan akan fasilitas yang dibutuhkan serta aktivitas apa saja yang dilakukan oleh sistem secara umum. Setelah informasi/data terkumpul, kemudian dilakukan penataan informasi dari kebutuhan pengguna tersebut, sehingga proses pengembangan perangkat lunak dapat terpenuhi.

4. *Product design solutions*

Pada tahapan ini dibuat solusi rancangan sistem dengan menggunakan Unified Modelling Language (UML) sesuai dengan kebutuhan pengguna.

5. *Evaluate design against user requirements*

Dalam tahap ini, evaluasi sistem terhadap perancangan yang disesuaikan dengan kebutuhan pengguna untuk melihat sejauh mana tujuan website telah tercapai.



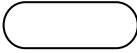



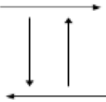
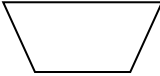

Gambar 2. 2 Metode UCD[10]

Adapun struktur data merupakan cara menyimpan atau mempresentasikan data didalam komputer agar bisa dipakai secara efisien. Berikut bagian-bagian yang ada pada struktur data:

A. *Flowchart*

Flowchart merupakan suatu diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Pada Tabel 2.1 merupakan simbol-simbol *flowchart*:

Tabel 2. 1 Simbol-simbol *flowchart*

No	Simbol	Nama	Keterangan
1		<i>Terminator Symbol</i>	Simbol yang menyatakan awal atau akhir suatu program.
2		<i>Input-Output</i>	Memasukkan data maupun menunjukkan hasil dari suatu proses tanpa tergantung dengan jenis pendataannya.
3		<i>Processing Symbol</i>	Menyatakan suatu proses yang dilakukan oleh komputer
4		<i>Decision symbol</i>	Menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya atau tidak.
5		<i>Flow symbol</i>	Menghubungkan antara simbol satu dengan simbol yang lain atau menyatakan jalannya arus dalam suatu proses. Simbol arus ini sering disebut juga dengan <i>connecting line</i> .
5		<i>Manual symbol</i>	Menyatakan suatu proses yang tidak dilakukan komputer
6		<i>Document Symbol</i>	Merupakan simbol untuk data yang terbentuk informasi.

B. *Unified Modeling Language (UML)*



Salah satu standar bahasa yang banyak digunakan didunia industry untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur disebut dengan *Unified Modeling Language (UML)*[16]. UML menyederhanakan permasalahan-permasalahan yang kompleks sehingga dapat dipelajari dan dipahami dengan mudah. UML dapat menggambarkan hubungan antar kelas dengan garis lurus.


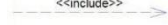



View-view dari UML terbagi menjadi 3 (tiga). *Structural classification*, *dynamic behavior* dan *model management*. *Structural classification* mendeskripsikan segala sesuatu dalam sistem dan relasi terhadap lainnya. *View* pengelolaan model mengorganisasikan sedemikian rupa untuk memudahkan pekerjaan pengembangan sistem seperti *class diagram*. Terdapat beberapa diagram UML yang digunakan dalam pengembangan sebuah sistem yaitu:

a. *Use Case Diagram*

Use case diagram merupakan permodelan untuk melakukan sistem informasi yang akan dibuat. *Use case diagram* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Tabel 2.2 menunjukkan simbol-simbol yang ada pada use case diagram[17].

Tabel 2. 2 Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1		<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
2		<i>Actor</i>	Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan <i>use case</i> .


3		<i>Assosiation</i>	Menghubungkan antara objek satu dengan objek lainnya.
4		<i>Include</i>	Menspesifikan bahwa <i>use case</i> sumber segala eksplisit.
5		<i>Extend</i>	Menspesifikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber suatu titik yang diberikan.
6		<i>System</i>	Menspesifikan paket yang menampilkan sistem secara terbatas.
7		<i>Generaliztion</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.

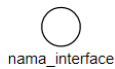





b. *Class Diagram*

Class diagram menunjukkan hubungan antar kelas dalam sebuah sistem yang akan dibangun. *Class diagram* ini digunakan untuk penggambaran struktur dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem[16]. Pada Tabel 2.3 menunjukkan simbol-simbol yang ada pada *class diagram*.

Berikut beberapa simbol dari *class diagram* :

Tabel 2. 3 Simbol *Class Diagram*


No	Simbol	Nama	Keterangan
1		<i>Class</i>	Kelas pada struktur sistem.

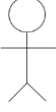




2		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3		<i>Association</i>	Relasi anatar kelas dengan makna umum.
4.		Coordinated associates	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
4		<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
5		<i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
6		<i>Aggregation</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

c. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan akan diterima antar objek[16]. *Sequence diagram* harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas. Tabel 2.4 berikut ini merupakan simbol-simbol yang ada pad *sequence diagram*.

Tabel 2. 4 Simbol *Sequence diagram*

No	Simbol	Nama	Keterangan
1		<i>Life line</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi

2		<i>Actor</i>	Menggambarkan hubungan yang akan dilakukan.
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.
4		<i>Boundary</i>	Menggambarkan sebuah <i>form</i> .
5		<i>Class control</i>	Menghubungkan <i>boundary</i> dengan tabel.
6		<i>Entity class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan.

Pengujian yang digunakan yaitu menggunakan metode BlackBox Testing merupakan sebuah metode yang digunakan untuk menguji sebuah software tanpa harus memperhatikan detail software. Proses BlackBox Testing dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada tiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan kebutuhan. Metode BlackBox Testing merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang diharapkan. Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid[14]. Pengujian BlackBox ini memiliki dua jenis pengujian yaitu pengujian fungsional dan pengujian non fungsional. Pengujian BlackBox (fungsionalitas) menguji bug hanya berdasarkan kegagalan fungsi perangkat lunak yang terungkap dalam bentuk output

yang salah. BlackBox Testing cenderung untuk menemukan hal-hal berikut:

1. Fungsi tidak benar atau tidak ada
2. Kesalahan antarmuka
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan perfomansi
5. Kesalahan inialisasi dan terminasi

2.2.3 Pemrograman Berorientasi Objek (PBO)

Pemrograman berorientasi objek merupakan pemrograman berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam *class-class* atau *object-object* [15]. Syarat sebuah bahasa pemrograman bisa digolongkan sebagai berorientasi objek adalah bila bahasa pemrograman tersebut memiliki fitur untuk mengimplementasikan ke 4 (empat) konsep berorientasi objek yaitu *abstraksi*, *encapsulation*, *polymorphison*, dan *inheritance*. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi.

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek. Metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi data dan fungsi. Sehingga, objek dapat memiliki kelakuan yang berbeda, apabila diperbarui implementasi dari objek tersebut sesuai dengan karakteristiknya.

2.2.4 Basis Data

Basis data merupakan koleksi dari data yang terorganisasi dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi. Disamping berisi atau menyimpan data, setiap basis data juga mengandung/menyimpan definisi struktur.

a) *Database Management System (DBMS)*

Database Management System (DBMS) atau dalam Bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. Dalam penggunaan DBMS dibutuhkan komponen-komponen antara lain[18]:

- (1) *Query processor*, komponen yang mengubah bentuk *query* dalam bentuk instruksi ke dalam *database manager*.
- (2) *Database manager*, menerima *query*, menguji eksternal dan konseptual untuk menentukan apakah *record-record* tersebut dibutuhkan untuk memenuhi permintaan kemudian hari dari *database manager* dengan memanggil *file manager* untuk menyelesaikan permintaan.
- (3) *File manager*, memanipulasi penyimpanan file dan mengatur alokasi ruang penyimpanan disk.
- (4) *Data manipulation language processor*, modul yang mengubah perintah DML yang ditempelkan ke dalam program aplikasi dalam bentuk fungsi-fungsi.
- (5) *Data definition language compiler*, mengubah statement DDL menjadi kumpulan tabel atau *file* yang berisi data *dictionary* atau meta data.
- (6) *Dictionary manager*, mengatur akses dan memelihara data *dictionary*.

Salah satu *software* yang tergolong ke dalam DBMS adalah *MySQL*. *MySQL* merupakan salah satu *Relational Database Management System* (RDBMS) yang saat ini sedang banyak diminati.

b) ***Structural Query Language (SQL)***

Structural Query Language (SQL) adalah Bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. Secara umum SQL terdiri dari 2 (dua) bahasa yaitu *Data Definition Language (DDL)* dan *Data Manipulation Language (DML)*.

(1) ***Data Definition Language (DDL)***

Data Definition Language (DDL) merupakan suatu perintah yang berfungsi untuk mendefinisikan atribut atribut basis data, tabel, atribut serta hubungan antar tabel. DDL berfungsi lebih ke dalam memanipulasi struktur *database*. DDL digunakan untuk membuat tabel atau menghapus tabel, membuat *key* atau *indeks*, membuat relasi antar tabel. Berikut sintaks yang ada didalam DDL:

(a) *Create*

Perintah *create* digunakan untuk membuat objek baru, baik berupa *database*, tabel, indeks atau prosedur yang tersimpan.

(b) *Alter*

Perintah *alter* digunakan untuk memodifikasi objek pada *database*, seperti *indeks*, dan lokasi.

(c) *Drop*

Perintah *drop* digunakan untuk menghilangkan atau menghapus objek pada *database*.

(2) **Data Manipulation Language (DML)**

Data Manipulation Language (DML) merupakan kelompok perintah yang berfungsi untuk melakukan proses insert, update atau delete ke dalam suatu *database*. Berikut sintaks yang ada didalam DML:

(a) *Select*

Perintah *select* digunakan untuk menampilkan data/isi tabel dari *database*.

(b) *Insert*

Perintah *insert* untuk menambahkan data baru dalam *database*.

(c) *Update*

Perintah *update* digunakan untuk merubah data didalam *database*.

(d) *Delete*

Perintah *delete* digunakan untuk menghapus data dari *database*.

2.2.5 Rumah Tidak Layak Huni (RTLH)

Rumah Tidak Layak Huni (RTLH) merupakan hunian yang tidak memenuhi standart keamanan dan kecukupan luas bangunan serta factor yang menjadi bahan pertimbangan dalam penentuan RTLH adalah melihat material yang digunakan dalam pembuatan rumah seperti dinding, lantai dan atap[1]. Suatu rumah dikatakan layak apabila memenuhi kriteria sesuai dengan kelayakan. Derajat kelayakan suatu rumah dapat diukur dari 2 (dua) aspek, yaitu kualitas fisik rumah dan kualitas fasilitas rumah. Kualitas fisik rumah dapat dilihat dari jenis atap, dinding rumah. Serta kualitas fasilitas rumah dilihat dari adanya jamban, ventilasi udara, sumber mata air yang memadai.

Halaman Sengaja Dikosongkan

