

BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Studi pustaka merupakan kajian dari buku, skripsi, majalah, jurnal, internet dan lain-lain yang digunakan sebagai referensi yang terkait dengan variabel penelitian yang relevan dengan hasil penelitian lain. Adapun penelitian serupa yang dapat diasumsikan memiliki relevansi dengan penelitian ini adalah :

Tabel 1 Penelitian terkait

No (1)	Nama (Tahun) (2)	Judul (3)
1	Prasetyoadi (2016) [6]	Judul : Sistem Kendali Suhu Dan Kelembaban Ruang Operasi Menggunakan Mikrokontroller Arduino Dan metode Logika Fuzzy Hasil Penelitian : Hasil penelitian ini adalah bahwa suhu ruangan dapat dideteksi menggunakan metode logika fuzzy tanpa memerlukan matematika dengan mengeluarkan output hanya menggunakan logika manusia. Seperti: dingin, sejuk, dan panas.
2	Irma n.d.(2019) [5]	Judul : Rancang Bangun Tampilan Seven Segment Pada Alat Baby Incubator Berbasis Mikrokontroler Atmega 89S52 Hasil Penelitian : Hasil penelitian ini ialah sensor dapat mendeteksi suhu dan kelembapan pada baby incubator dengan baik yang ditampilkan pada <i>7-segment</i> .
3	Dwiyanto (2020) [7]	Judul : Studi <i>Timer Digital Seven Segment</i> Berbasis Mikrokontroler At89s51 Hasil Penelitian : Hasil pengujian ini, seven segment menggunakan <i>software</i> Atmel Mikrokontroler ISP (<i>In-System Programming</i>) dapat berjalan dengan baik. Dimana angka atau nilai terbaca pada display <i>7-segment</i> .
4	Pratiwi Anggreini (2021) [3]	Judul : Penurunan Waktu Pra-Operasi Antar

	<p>Tindakan Pembedahan Menggunakan Analisa Changeover Time</p> <p>Hasil Penelitian : Hasil penelitian ini ialah dapat menghitung penurunan <i>Changeover time</i> pra operasi dengan menggunakan lampu menyala dan mati yang dapat diterapkan dengan efektif.</p>
5	<p>Hanim et al. (2020) [8]</p> <p>Judul : Rancang bangun start rangkaian buzzer pada alat infant warmer berbasis mikrokontroller atmega 8535</p> <p>Hasil Penelitian : Hasil penelitian ini ialah dapat membuat alat yang dapat digunakan untuk mendukung proses bertahannya bayi di luar kandungan. Alat ini menggunakan buzzer sebagai penandakan suhu dibawah batas suhu minimal, suhu diatas batas suhu maksimal, dan juga menandakan tegangan pada baterai dalam kondisi menurun.</p>
6	<p>Kharisma, Suprpto (2018) [9]</p> <p>Judul : Rancangan Penunjuk Waktu Auto Reload Menggunakan Gps Dan Rtc Dengan Modul Wifi Berbasis Arduino Di Poltekbang Surabaya</p> <p>Hasil Penelitian : hasil penelitian ini ialah RTC secara otomatis digunakan sebagai back up data waktu dan kalender yang apabila dari NTP terjadi kesalahan dalam penginputan dan GPS mati.</p>

Berdasarkan penelitian mengenai pewaktu, pendeteksi suhu dan kelembapan yang telah di sebutkan pada Tabel 1, penelitian ini membahas pewaktu pada ruang operasi dengan alarm sebagai penanda waktu telah habis dan dapat menampilkan suhu dan kelembapan pada ruang operasi dengan judul “Alarm Sistem Pewaktu Pada Ruang Operasi Rumah Sakit Menggunakan Penampil 7-Segment ”. Perangkat ini menggunakan platform NodeMCU ESP8266 dan sensor AHT10, yang dalam melakukan pengkonfigurasiannya menggunakan aplikasi berbasis *website* pada *handphone* yang terhubung dengan alat pewaktu operasi. Hasil penelitian ini dapat menghasilkan alat yang nantinya bisa diterapkan pada ruangan operasi, terutama pada rumah sakit yang saat ini masih menggunakan pewaktu manual.

B. Landasan Teori

1. Ruang Operasi

Rumah sakit menyediakan lingkungan yang aman dan nyaman untuk melakukan operasi baik untuk pasien maupun tenaga medis yang beraktifitas di dalamnya. Ruang operasi merupakan fasilitas yang minimal wajib dimiliki oleh rumah sakit dimana ruangan ini digunakan untuk tindakan pembedahan yang dilakukan oleh tim medis. Ruang operasi yang dimaksud bermakna sebagai suatu kompleks ruangan yang memiliki fasilitas atau ruangan di dalamnya untuk menunjang kelancaran dari tindakan pembedahan dan sterilisasi lingkungan [3]. Kenyamanan dan keamanan dalam ruang operasi ini dapat di capai dari dua hal yakni kenyamanan fisik dan kenyamanan non fisik. Salah satu kenyamanan fisik pada ruang operasi ialah Suhu kamar idealnya 20-26° C dan harus stabil dan Kelembaban ruangan 50-60%. Sedangkan kenyamanan non fisik salah satunya dapat dicapai dengan memberikan ruangan sesuai dengan kebutuhan kenyamanan hidup manusia dan mendesain ruangan agar bersuasana yang tidak membuat bosan [10].

2. *Timer*

Timer (Time Delay Relay) adalah sebuah komponen elektronik yang dibuat untuk menunda waktu yang bisa disetting sesuai *range timer* tersebut atau *Timer* adalah salah satu peralatan yang mempunyai fungsi sebagai pembatas waktu kerja suatu alat yang cara kerjanya berdasarkan sifat mekanis atau elektronis. Pada peralatan elektronis seperti televisi, audio set dan peralatan elektronik lainnya yang mempunyai fungsi pewaktu, pengaturan waktu kerjanya menggunakan mode *timer* dengan cara kerja yang bersifat elektronis. Pengaturan pembatasan waktu kerja tidak hanya diterapkan pada pembatasan waktu kerja suatu peralatan, tetapi juga diterapkan pada suatu kegiatan atau pekerjaan yang harus dilaksanakan dengan perioda waktu yang telah ditetapkan sebelumnya [4]. Nilai *timer* diatur secara otomatis ke nol setelah nilai maksimumnya. Setelah nilai maksimum untuk penghitung waktu tercapai, interupsi dihasilkan dengan *flag overflow*.

3. Aplikasi *Website*

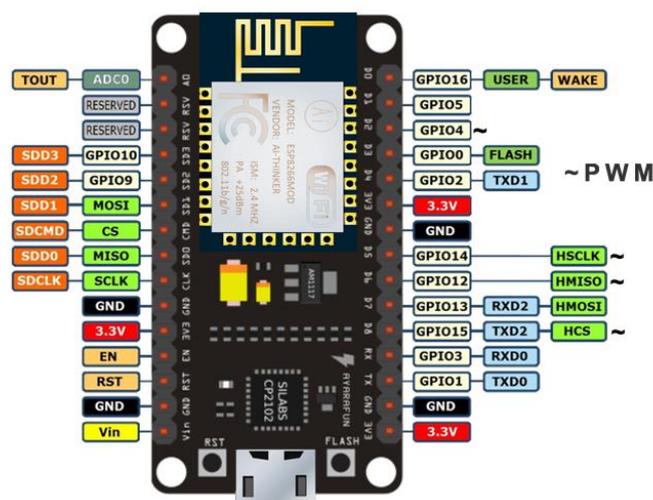
Aplikasi *website* adalah suatu bentuk aplikasi yang dapat di akses melalui *browser* tanpa memerlukan instalasi perangkat lunak tambahan di perangkat pengguna. Dengan kumpulan halaman-halaman informasi dalam bentuk data digital berupa teks, gambar, audio, video, dan animasi yang dapat diakses dengan menggunakan jaringan internet [11].

4. NodeMCU ESP8266

Menurut buku “Pemograman NodeMCU ESP8266 menggunakan Arduino IDE” bahwa NodeMCU adalah *development board* yang sudah ditanami chip ESP8266 sehingga tidak memerlukan perakitan komponen elektronika

sebagai pendukung sistem minimum mikrokontroler. Istilah "NodeMCU" secara default mengacu pada firmware dari pada kit pengembangan. Firmware menggunakan bahasa scripting Lua. Ini didasarkan pada proyek eLua, dan dibangun di atas Espressif Non-OS SDK untuk ESP8266. Secara fungsi modul ini hampir menyerupai dengan platform modul arduino, tetapi yang membedakan yaitu dikhususkan untuk "Connected to Internet". Pada NodeMCU didalam nya sudah dilengkapi dengan konverter USB ke Serial, fungsi dari USB ke Serial adalah protokol yang digunakan untuk mengubah sinyal data USB (*Universal Serial Bus*) ke atau dari standar komunikasi lainnya. atau dapat juga diartikan sebagai adaptor untuk mengubah data USB menjadi data port serial standar atau sebaliknya.

ESP8266 merupakan SoC (*System On Chip*) yang diproduksi oleh perusahaan *Espressif System Smart Connectivity Platform* (ESCP (*Espressif System*)) dan prosesor inti menggunakan 32bit Tensilicia L106. ESP8266 ditujukan untuk perancangan platform seluler dengan daya dan ruang terbatas namun ber-performa dan terintegrasi tinggi. yang menjadi daya tarik pada ESP8266 adalah mempunyai jaringan Wifi *independent*, yang berarti kita dapat menyimpan dan menjalankan aplikasi didalamnya tanpa dibutuhkan lagi prosesor eksternal (perangkat lainnya). Pada ESP8266 memiliki penyimpanan eksternal yang digunakan untuk menyimpan intruksi program (memori program). secara teori, ESP8266 dapat mendukung kapasitas memori sampai 16 MB dan kapasitas minimum yang di rekomendasikan adalah sebesar 1MB.



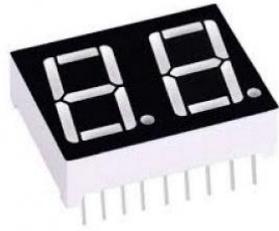
Gambar 1 NodeMCU ESP8266

PinOut pada NodeMCU ESP8266 terdiri dari 30 pin yang dapat digabungkan fungsinya menjadi 11 kelompok, antara lain :

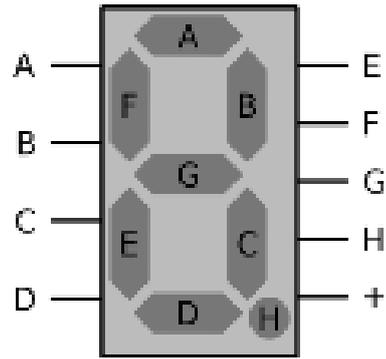
- a. Pin daya (power) pada NodeMCU terdiri dari empat pin, yaitu pin Vin dan tiga pin 3.3V. Pada pin Vin dapat digunakan catu daya eksternal seperti baterai dan catu daya adaptor. Tegangan harus antara 5V sampai 12V sehingga dapat menghidupkan NodeMCU dengan baterai eksternal 9V. Pin 3V dapat menghidupkan NodeMCU dengan tegangan 3.3V yang stabil pada pin 3.3V. Pin ini dapat digunakan sebagai pemasok daya eksternal dengan tegangan 3.3V, namun modul eksternal tidak boleh mengkonsumsi arus melebihi kemampuan regulator NodeMCU. Apabila voltasenya lebih tinggi dapat merusak NodeMCU. Mikro USB adalah cara paling mudah dalam menghidupkan NodeMCU ESP8266 dengan menggunakan kabel USB. Koneksi USB standar 5V.
- b. Pin GND adalah pin ground komponen eksternal yang mengambil daya dari NodeMCU, pin GND selalu menyertai dengan tegangan 3.3V.
- c. Pin I2C adalah pin yang digunakan untuk menghubungkan sensor dan peripheral I2C. Fungsionalitas antarmuka I2C dapat direalisasikan secara terprogram dan frekuensi maksimum adalah 100 kHz.
- d. Pin GPIO memiliki 17 pin GPIO yang dapat ditetapkan fungsinya seperti I2C, I2S, PWM, IR remote control, UART, lampu LED dan tombol secara terprogram. Pin GPIO juga dapat dikonfigurasi sebagai input yang dapat diatur ke edge-trigger atau level-trigger untuk menghasilkan interupsi CPU [12].

5. 7-Segment

7-segment adalah komponen elektronika yang dapat menampilkan angka ataupun nilai desimal melalui kombinasi-kombinasi segmentnya. Disebut *7-segment*, karena memiliki 7 batang LED yang membentuk angka 8. Dimana setiap *segment* dikendalikan on dan off agar dapat menampilkan nilai yang diinginkan. Angka-angka dari 0 – 9 dapat ditampilkan dengan menggunakan beberapa kombinasi segment. Selain 0 – 9, *7-segment display* juga bisa menampilkan huruf hexadecimal dari A – F. Pada *7-segment* ini diatur berbentuk angka “8” miring ke kanan, yang bertujuan untuk memudahkan dalam pembacaan angkanya. *7-segment* ini biasanya digunakan untuk menampilkan jam digital, kalkulator, pewaktuan dan kalender. Prinsip kerja dari *7-segment* ini adalah inputan bilangan biner pada switch dikonversi masuk ke dalam decoder, baru kemudian decoder mengkonversi bilangan biner tersebut ke dalam bilangan desimal, yang mana bilangan desimal ini akan ditampilkan pada layar *7-segment*. Fungsi dari decoder sendiri adalah sebagai pengkonversi bilangan biner ke dalam bilangan desimal [13].



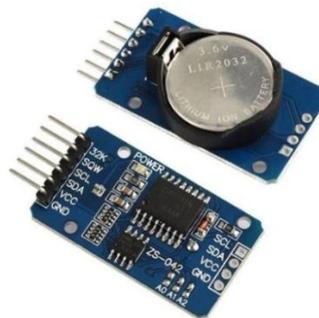
Gambar 2 7-Segment (a)



Gambar 3 Pin 7-Segment (b)

6. RTC DS3231

RTC merupakan alat yang digunakan untuk mengkases data informasi waktu dan kalender. RTC DS3231 mampu mengakses data waktu berupa jam, menit, detik, tanggal, bulan dan tahun yang bisa di sesuaikan secara otomatis. Pada DS3231 Operasi jam bisa diformat dalam 24 jam atau 12 jam (AM/ PM) dan setiap akhir bulan. Untuk tatap muka dengan suatu mikroprosesor dapat disederhanakan dengan menggunakan sinkronisasi komunikasi serial I2C dengan kecepatan clock 400Khz. Hanya membutuhkan 2 saluran untuk komunikasi dengan clock/RAM: SCL (serial clock), SDA (Serial I/O data), dan juga dilengkapi dengan keluaran SQW/Out yang dapat deprogram untuk mengetahui perubahan data waktu pada RTC dan pin RST [14]. Modul ini tidak dapat berdiri sendiri, melainkan merupakan suatu bagian dari sebuah perangkat ataupun rangkaian elektronik yang lebih besar atau lebih kompleks yang memiliki tugas khusus berkaitan dengan waktu dan tanggal. Tujuannya apabila rangkaian ataupun perangkat elektronik tersebut sewaktu-waktu mati sebab kehilangan sumber daya maupun dengan sengaja dimatikan dan dinyalakan secara berkala, sistem atau perangkat elektronik tersebut tetap bisa mendapatkan data waktu dan tanggal dengan akurat.



Gambar 4 RTC DS3231

7. Power Supply

Catu daya (*power supply*) adalah sebuah peralatan penyedia tegangan atau sumber daya untuk peralatan elektronika dengan prinsip mengubah tegangan listrik yang tersediadari jaringan distribusi transmisi listrik ke level yang diinginkan sehingga berimplikasi pada pengubahan daya listrik [15]. Fungsi dari *power supply* adalah merubah listrik AC (*Alternating Current*) menjadi tegangan DC (*Direct Current*). Pada *power supply* terdapat port dimana port ini digunakan untuk menghubungkan kabel *power* yang berarus AC akan melewati *power supply* yang di konversi menjadi arus DC. Fungsi dari *power supply* ini salah satunya menjaga kestabilan tegangan dan arus listrik meskipun terdapat perubahan atau variasi pada beban atau sumber listrik.



Gambar 5 Power Supply

8. Software Arduino IDE

IDE adalah singkatan dari *Integrated Development Environment*. Secara harfiah dapat diartikan sebagai lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Sehingga pengertian dari Arduino IDE (*Integrated Development Environment*) merupakan sebuah *software* yang digunakan untuk pengembangan dalam bentuk penulisan program, meng-*compile* menjadi kode biner dan mengunggah ke dalam memori mikrokontroler pada Arduino. Arduino IDE ini menggunakan bahasa pemrograman C++ dengan versi yang telah disederhanakan, sehingga menjadi lebih mudah dalam penggunaan. Program atau kode yang ditulis dalam Arduino IDE sering disebut sebagai sketch. Kita perlu menghubungkan board Arduino dengan Arduino IDE untuk mengunggah sketch yang ditulis dalam perangkat lunak Arduino IDE. Sketch disimpan dengan ekstensi '.ino' (Buku Pemrograman NodeMCU ESP8266 menggunakan Arduino IDE). Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-upload ke dalam memori mikrokontroler [16].

```

SurgeyTime | Arduino 1.8.20 Hourly Build 2021/12/20 07:33
File Edit Sketch Tools Help
SurgeyTime index.h
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include "index.h"
#include <EEPROM.h>
//=====
ACCESS POINT in SURGEY TIME
By: SEATONEDIA (SaFiX)
//=====
const char* ssid = "Surgey Time";
const char* password = "";
ESP8266WebServer server(80);
//=====
#include <AHT10.h>
AHT10 myAHT10(AHT10_ADDRESS_0X38);
float dataSuhu, dataKelembaban;
//=====
#define J_R1 D5 // GPIO14
#define J_ST D6 // GPIO12
#define J_SH D7 // GPIO13
#define Relay D4 // GPIO2
#define DIP D8 // GPIO15
boolean tandaDip;
int display, Temp_Harian, dataDigit, Data_Senuan, Data_Puluhan;
int humdS, humdP, suhuS, suhuP, timerSd, timerPd, timerSm;
int timerPm, timerSj, timerPj, tandaS, tandaP;
ESP8266 Dev Module, Disabled, Default 4MB with SPI (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi BT), QIO, 80MHz, 4MB (22MB), 921800, Core 1, Core 1, None on COM7

```

Gambar 6 Software Arduino IDE

9. Sensor AHT10

Sensor AHT10 merupakan sensor alternatif sensor suhu dan kelembaban selain sensor DHT11 atau LM35. Sensor AHT10 merupakan komunikasi i2c dengan harga yang terjangkau dan presisi. Sehingga pengkabelan pada sensor ini lebih simple dan tingkat pembacaan datanya lebih stabil. Sensor ini menetapkan standar baru dalam ukuran dan kecerdasan yang tertanam dalam paket SMD tanpa timbal datar dua baris untuk penyolderan *reflow* dengan bagian bawah 4x5mm dan ketinggian 1.6mm. Sensor AHT10 sudah dilengkapi dengan chip asic-spesifik yang baru dirancang, elemen penginderaan kelembaban kapasitif semikonduktor MEMS yang ditingkatkan dan elemen penginderaan suhu on-chip standar. Kinerjanya telah sangat meningkat melampaui tingkat keandalan sensor generasi sebelumnya. Generasi pertama sensor suhu dan kelembaban telah ditingkatkan untuk membuatnya lebih stabil pada semua lingkungan [17].



Gambar 7 Sensor AHT10

10. UML (*Unified Modeling Language*)

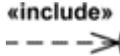
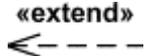
UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan

melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek. UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya [18].

a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case* [18].

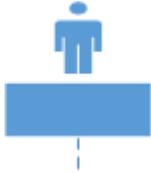
Tabel 2 Simbol *Use Case Diagram*

No	Simbol	Nama Simbol dan Keterangan
(1)	(2)	(3)
1.		Actor adalah menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		Use Case menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu.
3.		Association penghubung antara <i>Actor</i> dan <i>Use Case</i>
4.		Include yaitu perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
5.		Extend yaitu perilaku suatu <i>use case</i> memperluas perilaku <i>use case</i> lain. <i>Use case</i> boleh dilakukan boleh tidak (optional)
6.		Sistem Boundary yaitu batasan dari sebuah sistem.

b. *Sequence Diagram*

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Dalam diagram *sequence*, setiap object hanya memiliki garis yang digambarkan garis putus-putus ke bawah. Pesan antar object digambarkan dengan anak panah dari object yang mengirimkan pesan ke object yang menerima pesan [19].

Tabel 3 Simbol *Sequence Diagram*

No	Simbol	Keterangan
(1)	(2)	(3)
1.		<i>aktor</i> , orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri
2.		Garis hidup/ <i>lifeline</i> , menyatakan kehidupan suatu objek
3.		Objek, menyatakan objek yang berinteraksi pesan
4.		Waktu aktif, menyatakan objek dalam keadaan aktif dan berinteraksi
5.		Pesan tipe <i>create</i> , menyatakan suatu objek membuat objek lain arah panah mengarah pada objek yang dibuat
6.		Pesan tipe <i>call</i> , menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri

7.		Pesan tipe <i>send</i> , menyatakan bahwa suatu objek mengirim data atau masukan atau informasi ke objek lainnya arah panah mengarah pada objek yang dikirim
8.		Pesan tipe <i>return</i> , menyatakan bahwa suatu objek yang telah menjalankan operasi atau metode menghasilkan suatu kembalian ke objek tertentu arah panah mengarah pada objek yang menerima kembalian
9.		Pesan tipe <i>destroy</i> , menyatakan suatu objek mengakhiri hidup objek yang lain arah panah mengarah pada objek yang diakhiri sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

c. Activity Diagram

Activity Diagram merupakan salah satu jenis diagram yang digunakan dalam pengembangan perangkat lunak untuk menggambarkan alur dan urutan aktivitas dalam sebuah sistem [20]. Berikut ini symbol-simbol yang terdapat pada *activity diagram* :

Tabel 4 Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
(1)	(2)	(3)	(4)
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

d. Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari

pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class* diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek. Yng dikelompokkan ke dalam tiga area pokok yaitu nama, atribut dan metode [21]. Atribut dan metoda dapat memiliki salah satu sifat berikut:

- 1) *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
- 2) *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.
- 3) *Public*, dapat dipanggil oleh siapa saja.

Berikut adalah beberapa simbol-simbol yang ada pada *class* diagram dapat dilihat pada tabel 4.

Tabel 5 Simbol-simbol *Class Diagram*

No	Simbol	Nama dan Keterangan
(1)	(2)	(3)
1.		<i>Asosiasi</i> adalah hubungan antar kelas.
2.		<i>Generalization</i> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus).
3.		<i>Composition</i> adalah bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yg menjadi part ikut musnah.

e. *Prototype*

Prototype adalah model atau simulasi dari semua aspek produk sesungguhnya yang akan dikembangkan, model ini harus bersifat representatif dari produk akhirnya. Pada pengembangan sistem seringkali terjadi keadaan dimana pengguna sistem sebenarnya telah mendefinisikan secara umum atau tujuan perangkat lunaknya meskipun belum mendefinisikan secara rinci masukan, proses dan keluaran [20]. Perancangan *prototype* memiliki tujuan untuk mengembangkan rancangan atau model produk sampai menjadi produk final yang bisa memenuhi kebutuhan serta keinginan user. Manfaat dari *prototype* ini antara lain adalah lebih mudah dalam mempresentasikan produk dan menjadi acuan dalam pengembangan produk.