

BAB II TINJAUAN PUSTAKA

2.1 Referensi Jurnal

2.1.1 Aplikasi Presensi Mahasiswa Berbasis Web

Penelitian yang dilakukan Hendri Ariyanto (2016) pada skripsi Aplikasi Presensi Mahasiswa Berbasis Web, kriteria yang digunakan untuk proses presensi yaitu mahasiswa dari suatu Universitas atau Perguruan Tinggi melakukan presensi dengan memasukkan NIM dan Nama pada form presensi berbasis web, sehingga memberikan kemudahan bagi seorang dosen untuk melihat dan mengecek kehadiran mahasiswa secara *real time*, data per kelas, cetak laporan kegiatan presensi berdasarkan kelas atau mata kuliah tertentu secara online.

2.1.2 Aplikasi Pencatatan Kehadiran Karyawan Menggunakan Komputer Dengan Barcode Id Card Dan Pendeteksi Jari

Penelitian yang dilakukan Darwin Salim (2015) dalam skripsi Aplikasi Pencatatan Kehadiran Karyawan Menggunakan Komputer dengan *Barcode Id Card* dan Pendeteksi Jari, kriteria yang digunakan yaitu Karyawan dari suatu perusahaan dengan melihat kuantitas kerja, kualitas kerja, pemanfaatan waktu, kerjasama, kehadiran, jarak dari perusahaan sehingga dapat meningkatkan presensi atau tingkat kehadiran karyawan.

2.1.3 Aplikasi Presensi Kepegawaian Kelurahan Menggunakan Teknologi Fingerprint

Penelitian yang dilakukan Dhanni Wibawa (2016) dalam skripsi Aplikasi Presensi Kepegawaian Kelurahan menggunakan teknologi *fingerprint*, kriteria yang digunakan yaitu Pegawai dari kelurahan, dengan melakukan pendaftaran atau mengisikan biodata pada aplikasi dengan teknologi *fingerprint*/sidik jari, selanjutnya Sistem Presensi pegawai dengan mesin sidik jari dapat melakukan absensi pegawai kelurahan untuk harian maupun bulanan.

2.1.4 Sistem Absensi Mahasiswa Menggunakan Fingerprint Pada Asrama Politeknik Sekayu Berbasis Php dan Mysql

Penelitian yang dilakukan Agus Setiawan (2013) dalam Tugas Akhir Sistem Absensi Mahasiswa Menggunakan Fingerprint Pada Asrama Politeknik Sekayu Berbasis Php dan Mysql, Membangun sebuah sistem pengolah data absensi dengan menggunakan fingerprint sebagai pencatat kehadiran mahasiswa dan memberikan kemudahan dalam proses absensi bagi para mahasiswa dan meningkatkan efisiensi waktu dalam membuat laporan kehadiran mahasiswa pada Asrama Politeknik Sekayu Musi Banyuasin.

2.1.5 Sistem Absensi Mahasiswa Menggunakan Fingerprint dan Visual Basic 6.0

Penelitian yang dilakukan Chintya Bunga Yudhitiara (2015) pada skripsi Sistem Absensi Mahasiswa Menggunakan *Fingerprint* dan *Visual Basic 6.0*, kriteria yang digunakan untuk proses absensi yaitu mahasiswa dari suatu Universitas atau Perguruan Tinggi melakukan presensi dengan menggunakan *fingerprint* dengan perancangan antarmuka tampilan menggunakan *Visual Basic 6.0*, Mempermudah dosen dalam pencarian data mahasiswa dan sebagai wahana untuk meningkatkan kedisiplinan mahasiswa dalam mengikuti tatap muka perkuliahan.

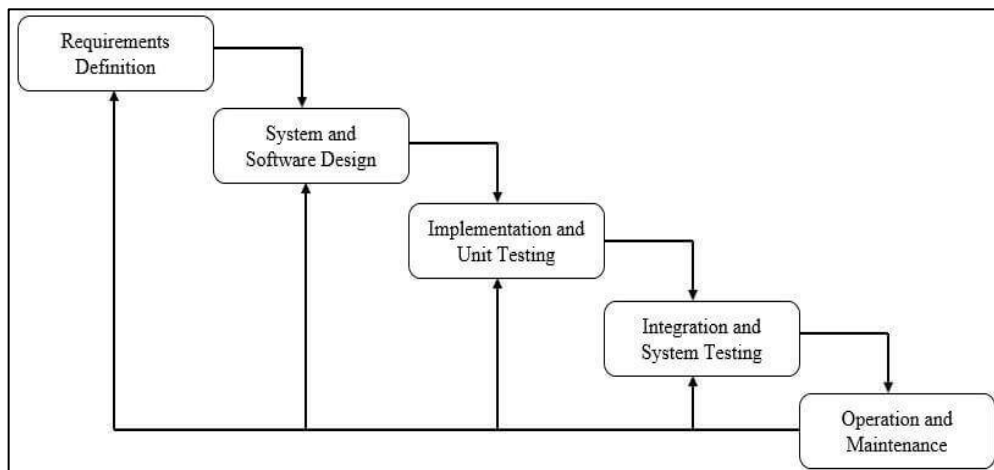
Sedangkan penelitian yang dilakukan peneliti tidak jauh berbeda dengan penelitian sebelumnya yaitu merancang website presensi karyawan berbasis gps untuk suatu instansi atau lembaga. Website ini menggunakan metode waterfall yaitu metode pengembangan perangkat yang berurutan sehingga tidak terjadi pengulangan pengumpulan data, analisa sistem, dan perancangan sistem serta pembangunan sistem yang berulang sehingga waktu yang digunakan lebih efisien.

2.2 Metode Pengembangan

2.2.1 System Development Life Cycle (SDLC)

SDLC adalah tahapan-tahapan pekerjaan yang dilakukan oleh analis sistem dan *programmer* dalam membangun sistem informasi sehingga dapat digunakan oleh pengguna sebagai pendukung kebutuhan bisnis. Salah satu SDLC yang paling sering digunakan dalam pengembangan sistem yaitu SDLC *Waterfall*. Rosa (dalam Hartini, 2017) model *Waterfall* sering juga disebut model (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*).

2.2.2 Tahapan Metode Waterfall



Gambar 2.1 Tahapan Metode Waterfall

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasikan kebutuhan perangkat lunak agar dapat dipahami

perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

5. Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.3 Landasan Teori

2.3.1 Pengertian Presensi

Presensi adalah pencatatan dan pengolahan data kehadiran yang dilakukan secara terus menerus, pencatatan dilakukan setiap hari jam kerja dan dilakukan pelaporan kepada pihak HRD / Manager Perusahaan. (Redi Mulyana, 2017).

2.3.2 Website

Website adalah sebuah software yang berfungsi untuk menampilkan dokumen pada suatu web yang membuat pengguna dapat mengakses *internet* melalui *software* yang terkoneksi dengan *internet*. (Mara Destiningrum, Q. All, 2017).

2.3.3 Bootstrap

Bootstrap adalah platform untuk membuat interface website dan aplikasi berbasis web. *Bootstrap* berisi kode *HTML* dan *CSS* yang telah dilengkapi desain untuk tipografi, bentuk, tombol, navigasi dan sebagainya. *Bootstrap* bertujuan untuk meringankan pembuatan dan pengembangan web. Dapat Disimpulkan bahwa *bootstrap* merupakan sebuah aplikasi yang dijadikan sebagai alat bantu untuk mendesain halaman web yang menarik. (Sitinjak Daniel Dido Jantce TJ & Suwita, 2020).

2.3.4 Codeigniter

Codeigniter merupakan *framework PHP* yaitu yang bersifat *open source* sehingga dapat digunakan oleh siapapun dan mengadopsi pola desain *MVC* (*Model, View, Controller*) yang memungkinkan pengembangan *website* secara dinamis dan memungkinkan penemuan serta penanganan *error* yang lebih cepat.

Penggunaan *framework CodeIgniter* tidak hanya membantu memudahkan proses pengkodean tetapi juga membuat *website* yang akan dibangun menjadi lebih aman karena memiliki tingkat *sekuritas* yang lebih tinggi dibanding *framework PHP* lain. (Solanki, Shah, & Shah, 2017).

2.3.5 PHP (*Hypertext Pre-processor*)

Biasa disebut singkat dari *Hypertext Pre-processor*, *PHP* adalah Bahasa pemrograman umum yang dapat disematkan kedalam kode *HTML* yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi Web yang dapat diubah tampilan dan isinya sehingga memberikan *feedback* kepada *user*. (Mundzir MF, 2018).

2.3.6 CSS (*Cascading Style Sheet*)

CSS adalah suatu Bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam. *CSS* merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. (Wahyudi, 2017).

2.3.7 HTML (*Hypertext Markup Language*)

HTML merupakan singkatan dari Hypertext Markup Language yang digunakan untuk membuat struktur dalam mendesain sebuah halaman *Website* yang bisa digunakan untuk menulis teks, memasukkan gambar, membuat form, dan lain-lain. *HTML* ini digabungkan dengan *CSS* dan *script* lain seperti *javasscript*, hal tersebut terjadi karena *HTML* tidak bisa berdiri sendiri. (Jubilee Enterprise, 2018).

2.3.8 HTTP (*Hypertext Transfer Protocol*)

HTTP merupakan singkatan dari (*Hypertext Transfer Protocol*) *HTTP* ini merupakan sebuah protokol yang bertujuan agar *client* dan *server* bisa berkomunikasi dengan gaya *request-response*. *HTTP* menentukan bagaimana *webserver* dan *browser* bereaksi dan bereaksi terhadap berbagai perintah, juga menentukan bagaimana format pesan dan cara pengirimannya. (Hidayatullah & Jauhari Khairul Kawistra, 2017).

2.3.9 MySQL (*My Structured Query Language*)

MySQL digunakan dalam mengelola dan membuat sebuah *database*, yang didalamnya mempunyai program khusus atau perintah dasar yaitu *query SQL (Structured Query Language)*. Sebelum data disimpan dalam *database MySQL*, *database* ini digunakan dalam pengimputan data menggunakan *HTML*, setelah itu diolah *PHP* yang kemudian dapat tersimpan pada *database MySQL*. (Jubilee Enterprise, 2018).

2.3.10 XAMPP

XAMPP adalah perangkat lunak (*free software*) bebas, yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. Fungsi *XAMPP* sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. (Mawaddah dan Fauzi, 2018).

2.3.11 PHP MyAdmin

PHP MyAdmin adalah suatu aplikasi *open source* yang berbasis web, aplikasi ini dibuat menggunakan program *PHP*, dan berfungsi untuk mengakses database *MySQL*. (Mulyanto & Khasanah, 2018).

2.3.12 Basis Data

Basis Data merupakan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi sehingga dapat disimpulkan bahwa basis data merupakan data yang dapat didesain dan berintegrasi agar dapat memenuhi kebutuhan user dalam perusahaan atau organisasi. (Pamungkas, 2017:6).

2.3.13 Sistem Basis Data

Sistem Basis Data adalah suatu sistem menyusun dan mengelola record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan. (Nurhadi dkk, 2019:6).

2.3.14 GPS (*Global Positioning System*)

GPS (Global Positioning System) adalah suatu sistem navigasi yang menggunakan satelit dan komputer untuk menentukan posisi lintang dan bujur dari suatu *client* dengan cara melakukan komputasi terhadap perbedaan waktu sampainya sinyal dari beberapa satelit yang berbeda ke *user*. (Diah Chaerani & Erick Paulus, 2017:54).

2.3.15 Pengertian Koordinat

Koordinat : bilangan yang dipakai untuk menunjukkan lokasi suatu titik dalam garis, permukaan, atau ruang. Koordinat adalah suatu titik yang didapatkan dari hasil perpotongan dari garis latitude (lintang) dengan garis bujur (longitude) sehingga akan menunjukkan lokasi pada suatu daerah. Umumnya koordinat dibedakan menjadi koordinat Geographic dan Universal Transver Mercator (UTM). Pada Koordinat Geographic dibedakan menjadi tiga berdasarkan satuannya yaitu :

1. Degree, Decimal (DD,DDDD) Contoh : S 3.56734 E 104.67235
2. Degree, Minute (DD MM,MMMM) Contoh : S 3°43,5423' E 104 33,6445'
3. Degree, Minute, Second (DD MM SS,SS) Contoh : S 3°43' 45,22" E104 33' 33,25"

Pada Bujur/Longitude (X) merupakan garis yang perpindahannya secara vertical dan pada Lintang/Lattitude (Y) merupakan garis yang mempunyai perpindahan secara horizontal, pada (Gambar 1) menjelaskan perpotongan antara garis bujur dan garis lintang akan membentuk suatu titik pertemuan yang biasa disebut dengan titik koordinat.

2.3.15.1 Jenis Titik Koordinat

1. Latitude

Latitude atau Garis Lintang adalah garis yang menentukan jarak di sebelah utara atau selatan Khatulistiwa. Latitude atau garis lintang diukur mulai dari 0 derajat dari Khatulistiwa dan berakhir pada 90 derajat di kutub (selatan dan utara). Dengan kata lain, Garis Lintang atau Latitude adalah suatu cara numerikal yang mengukur seberapa jauh utara atau selatan dari garis Khatulistiwa. Garis Khatulistiwa adalah

titik awal untuk mengukur Garis Lintang. Oleh karena itu, Garis Khatulistiwa biasanya ditandai sebagai 0 derajat.

2. Longitude

Longitude atau Garis Bujur adalah garis yang membentang dari utara ke selatan. Garis Bujur atau Longitude ini biasanya juga disebut dengan garis Meridian. Garis Bujur dapat juga dikatakan sebagai cara numerikal yang mengukur atau menunjukkan seberapa jauh lokasi timur atau barat dari garis vertikal universal yaitu garis Meridian Utama (Prime Meridian). Seperti yang disepakati oleh dunia Internasional, garis Meridian Utama berada tepat di atas British Royal Observatory di Greenwich Inggris, dari Kutub Utara ke Kutub Selatan. Sebagai titik awal vertikal untuk garis bujur, Meridian Utama diberi nomor garis bujur 0 derajat.

3. Altitude

Altitude (atau elevasi) adalah posisi vertikal (ketinggian) suatu objek dari suatu titik tertentu (datum). Datum yang biasa digunakan adalah permukaan laut dan permukaan geoid WGS-84 yang digunakan oleh GPS. Oleh karena itu, altitudo sering kali dinyatakan sebagai ketinggian dari permukaan laut (biasa disingkat dpl).

2.3.15.2 Membaca Dan Menghitung Koordinat

1. Membaca koordinat

Cara membaca latitude dan longitude angka koordinat lokasi bumi dibagi menjadi 3 format, koordinat di Google Maps berbeda dengan dengan koordinat di GPS (Global Positioning System). Namun pada beberapa GPS sudah mencantumkan ketiga jenis koordinat. Berikut ini adalah ketiga jenis koordinat :

- Derajat, menit, dan detik (D°M'S'")
Koordinat ini paling umum digunakan untuk GPS dengan akurasi.
Contoh koordinat lokasi bumi dengan D°M'S'": 7°48'10.2"S
110°22'26.9"E
- Derajat dan menit desimal (DMM)
Derajat, menit, desimal. Paling umum digunakan pada perangkat elektronik. Contoh : 41 24.2028, 2 10.441. Angka 41 misalnya adalah derajat, dan angka setelah 41 adalah angka dari pembagian 60 dari koordinat desimal.
- Derajat desimal (DD°): -7.802845, 110.374136
Derajat. Digunakan untuk koordinat komputer. Paling sederhana dengan memasukan 2 angka koordinat decimal -7.802845, 110.374136.

2. Menghitung Koordinat

Cara menghitung latitude dan longitude :

- Contoh perhitungan koodinat D°M'S' (GPS) ke kordinat DD°
Jika pada layar GPS tertera : 7°48'10.2"S , 110°22'26.9"E, artinya :
S (South / Lintang Selatan / Latitude) : 7 derajat 48 menit 10.2 detik
E (Bujur / Longitude) : 110 derajat 22 menit 26.9 detik.

- Rumus :

Koordinat desimal = derajat + (menit/60) + (detik/3600) maka

$$S = 7 + (48/60) + (10.2/3600) = 7.8028333$$

$$E = 110 + (22/60) + (26.9/3600) = 110.374138$$

Jangan lupa untuk menambahkan minus (-) pada koordinat latitude untuk lokasi yang berada di sebelah selatan garis equator (karena tertulis S atau South atau Lintang Selatan). Jadi, koordinat desimal yang didapat adalah:

Latitude (Lintang) : -7.8028333 Longitude (Bujur) : 110.374138

2.3.16 Pengertian Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan flowchart akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu flowchart juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah. (Santoso & Radna Nurmawati, 2017).

2.3.16.1 Jenis Flowchart

Flowchart sendiri terdiri dari lima jenis, yaitu :

1. Flowchart dokumen

Flowchart dokumen (document flowchart) atau bisa juga disebut dengan paperwork flowchart. Flowchart dokumen berfungsi untuk menelusuri alur form dari satu bagian ke bagian yang lain, termasuk bagaimana laporan diproses, dicatat, dan disimpan.

2. Flowchart program

Flowchart program. Flowchart ini menggambarkan secara rinci prosedur dari proses program. Flowchart program terdiri dari dua macam, antara lain: flowchart logika program (program logic flowchart) dan flowchart program komputer terinci (detailed computer program flowchart).

3. Flowchart proses

Flowchart proses adalah cara penggambaran rekayasa industrial dengan cara merinci dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

4. Flowchart sistem

Flowchart sistem adalah flowchart yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh. Selain itu flowchart sistem juga menguraikan urutan dari setiap prosedur yang ada di dalam sistem.

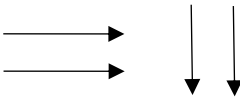
5. Flowchart skematik

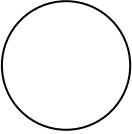
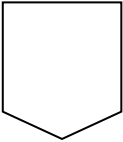
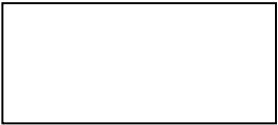
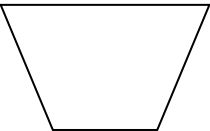
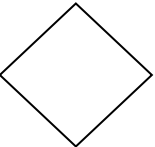
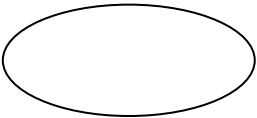
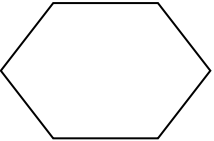
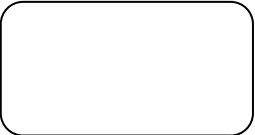
Flowchart ini menampilkan alur prosedur suatu sistem, hampir sama dengan flowchart sistem. Namun, ada perbedaan dalam penggunaan simbol-simbol dalam menggambarkan alur. Selain simbol-simbol, flowchart skematik juga menggunakan gambar-gambar komputer serta peralatan lainnya untuk mempermudah dalam pembacaan flowchart.

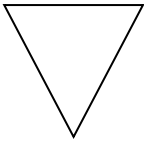
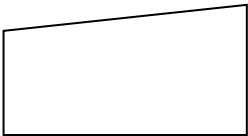

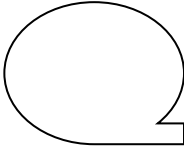




2.3.16.2 Simbol Flowchart

Pada dasarnya simbol-simbol dalam flowchart memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan flowchart.

Tabel 2.1 Simbol diagram *flowchart*

NO	SIMBOL	KETERANGAN
1		Simbol arus/ <i>flow</i> (Simbol penghubung antar prosedur/proses)

2		Simbol <i>connector</i> (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang sama)
3		Simbol <i>off-line connector</i> (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang lain)
4		Simbol <i>process</i> (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)
5		Simbol <i>manual operation</i> (Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer)
6		Simbol <i>decision</i> (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi)
7		Simbol <i>terminal</i> (Simbol untuk permulaan atau akhir dari suatu program)
8		Simbol <i>predefined process</i> (Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i>)
9		Simbol <i>keying operation</i> (Simbol operasi dengan menggunakan mesin yang mempunyai <i>keyboard</i>)

10		Simbol <i>off-line storage</i> (Simbol yang menunjukkan bahwa data di dalam simbol ini akan disimpan)
11		Simbol <i>manual input</i> (Simbol untuk pemasukan data secara <i>manual on-line keyboard</i>)
12		Simbol <i>input-output</i> (Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya)
13		Simbol <i>magnetic-tape unit</i> (Simbol yang menyatakan <i>input</i> berasal dari pita magnetik atau <i>output</i> disimpan ke pita magnetik)
14		Simbol <i>disk and on-line storage</i> (Simbol untuk menyatakan <i>input</i> berasal dari disk atau <i>output</i> disimpan ke disk)
15		Simbol <i>document</i> (Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak ke kertas)
16		Simbol <i>punched card</i> (Simbol yang menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu)
17		Simbol <i>display</i> (Simbol yang menyatakan peralatan <i>output</i> yang

		digunakan yaitu layar, <i>plotter</i> , printer, dan sebagainya)
--	--	--

2.3.16.3 Aturan Menggambarkan Flowchart

Adapun aturan dalam menggambar Flowchart antara lain :

1. Pertama mempertimbangkan logika utama, kemudian menggabungkan rincian.
2. Konsisten mempertahankan tingkat detail untuk flowchart.
3. Tidak perlu mencakup semua rincian dalam flowchart.
4. Pengguna deskripsi yang berarti dalam simbol-simbol diagram alur. Ini harus mudah dipahami.
5. Konsisten dalam menggunakan variabel dan nama dalam flowchart.
6. Untuk flowchart kompleks, konektor pengguna untuk mengurangi jumlah garis aliran. Persimpangan dari garis harus dihindari sejauh mungkin.
7. Jika flowchart tidak dapat digambarkan pada satu halaman, dianjurkan untuk membagi sesuai input atau output dan berikan label konektor dengan benar untuk menghubungkan bagian-bagian dari flowchart pada halaman terpisah.
8. Hindari duplikasi sejauh mungkin.
9. Aliran flowchart, harus dari atas ke bawah dan dari kiri ke kanan.
10. Awal dan akhir flowchart harus jelas. Tunjukkan dari mana asal dokumen dan tunjukkan posisi akhir dokumen sehingga tidak ada aliran tidak berujung yang membingungkan pembaca.
11. Gunakan simbol anotasi jika deskripsi input/proses/output tidak cukup dituliskan di dalam masing-masing simbol.

12. Dokumen yang memiliki salinan, tuliskan nomor dokumen tepat di pojok kanan atas setiap simbol.
13. Simbol proses manual, harus memiliki input dan output. Untuk simbol dokumen, tidak boleh langsung berhubungan dengan dokumen, kecuali saat dokumen berpindah ke kolom lain.
14. Tampilkan dokumen pada kolom sumbernya.
15. Arsip. Data dari dan menuju file komputer, harus melalui simbol proses terlebih dahulu. Namun untuk dari dokumen ke file, gunakan garis karena proses manual memasukkan dokumen menjadi file tidak perlu ditampilkan.
16. Gunakan off-page connector untuk berpindah dari satu halaman flowchart ke halaman lain.
17. Gunakan panah pada semua aliran garis.

2.3.17 Microsoft Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi *editor code open source* yang dikembangkan oleh *Microsoft* untuk sistem operasi *Windows*, *Linux*, dan *MacOS*. *Visual Code* memudahkan dalam penulisan kode yang mendukung beberapa jenis pemrograman, seperti *C++*, *C#*, *Java*, *Python*, *PHP*, *GO*. *Visual Code* memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian kode tersebut. *Visual Studio Code* juga telah terintegrasi ke *Github*. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code*. (Nur Ariesanto Ramdhan, 2019).