



BAB II
TINJAUAN PUSTAKA DAN
LANDASAN TEORI

BAB II

TEORI PENUNJANG

2.1 Tinjauan Pustaka

Beberapa penelitian sebelumnya yang dilakukan oleh penulis lain yang dijadikan sebagai bahan pertimbangan yang berkaitan dengan penelitian yang pertama adalah dilakukan oleh [3] membahas tentang pegawai yang ingin melakukan cuti tahunan, cuti melahirkan, izin dinas, dan izin sakit masih menggunakan sistem manual, yaitu dengan cara pegawai mendatangi *staff* absensi dengan membawa surat dokumen ketidakhadiran yang nanti akan diinput data absensi ketidakhadiran oleh *staff* absensi. Sistem ini dirancang dengan menggunakan perangkat lunak *Waterfall* dengan menggunakan bahasa pemrograman *PHP* dan *MySQL*. Pengujian *web* dilakukan dengan menggunakan *blackbox*. Observasi, wawancara dan daftar pustaka juga digunakan dalam teknik pengumpulan data. Dalam sistem informasi yang dirancang dua pengguna yang dapat saling berinteraksi dalam lingkup sistem, yaitu admin dan pegawai.

Penelitian serupa juga pernah dilakukan oleh [4] membahas pada Peraturan Daerah Kota Tangerang Selatan Nomor 61 Tahun 2016 tentang Kedudukan, Struktur Organisasi, Tugas, Fungsi dan Tata Kerja dibentuklah Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP). Awal mula terbentuknya DPMPTSP dimana sistem masih berjalan manual dan masih terdapat praktek pungli dilakukan oknum calo. Sistem ini dirancang dengan menggunakan metode pengumpulan data, metode konsep analisa sistem menggunakan *SDLC (Sistem Development Life Cycle)* tipe *waterfall* dan metode rancangan sistem menggunakan *UML (Unified Modeling Language)* dan *CI Framework*.

Penelitian serupa juga pernah dilakukan oleh [5] membahas tentang *staff* karyawan dan dosen yang akan mengajukan permohonan cuti harus menyampaikan pengajuan tersebut secara tertulis menggunakan formulir permohonan cuti. Penerapan cara manual ini membutuhkan waktu yang lama karena belum terkomputerisasi. Selain itu, kurang efektif dalam proses kerjanya, karena banyaknya pemakaian kertas sehingga sulit dalam pengontrolan persetujuan permohonan. Sistem ini dirancang dengan metode *Rapid Application Development*

(RAD). Terdapat tiga fase dalam RAD yaitu *planning, design workshop dan implementation*.

Berbeda dengan penelitian-penelitian terkait yang dilakukan sebelumnya. Adapun penelitian yang dikembangkan berjudul Sistem Informasi Permohonan Izin Mahasiswa Berbasis *Website* pada Jurusan Komputer dan Bisnis Politeknik Negeri Cilacap menggunakan *PHP Native* dengan metode pengembangan *prototype*, yang nantinya akan digunakan oleh mahasiswa, dosen wali, ketua jurusan, akademik dan admin jurusan. Sehingga dapat mempermudah mahasiswa dalam melakukan permohonan izin dan dapat melihat riwayat setiap semester, mempermudah dosen wali memantau mahasiswa melalui perantara internet, mempermudah bagian jurusan dan akademik dalam melakukan pengecekan data. Dalam sistem ini dilengkapi dengan fitur notifikasi *via Email*. Dosen wali, ketua jurusan dan akademik mendapatkan notifikasi jika ada mahasiswa yang melakukan izin dan peringatan rekomendasi SP I jika mahasiswa alpha lebih dari sama dengan 10 jam, SP II lebih dari sama dengan 20 jam dan SP III lebih dari sama dengan 30 jam.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem yang dibuat oleh manusia, yang terdiri dari komponen - komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi menerima masukan data dan instruksi, mengolah data tersebut sesuai instruksi, dan mengeluarkan hasilnya. Model dasar sistem menghendaki agar masukan, pengolahan, dan keluaran tiba pada saat bersamaan, yang sebaiknya sesuai untuk sistem pengolahan informasi yang paling sederhana, di mana semua masukan tersebut tiba pada saat bersamaan, meskipun hal tersebut jarang terjadi.

Fungsi pengolahan informasi sering membutuhkan data yang telah dikumpulkan dan diolah dalam periode waktu sebelumnya. Oleh karena itu ditambahkan sebuah penyimpanan data *file* (*data file storage*) ke dalam model sistem informasi. Dengan demikian, kegiatan pengolahan

tersedia baik bagi data baru maupun data yang telah dikumpulkan dan disimpan sebelumnya.[6]

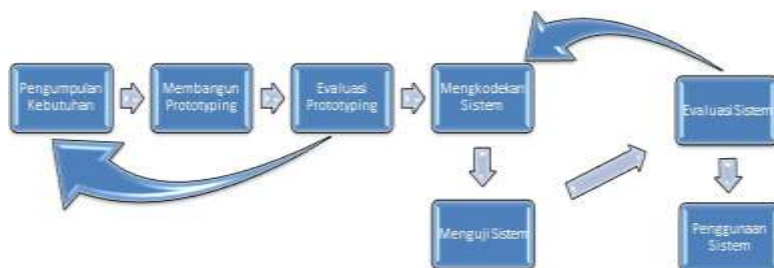
2.2.2 Rekayasa Web

Rekayasa *Web* merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin.

Berikut adalah metode dan *tools* yang digunakan :

1. Metode[7]

Pengembangan sistem dalam pembangunan aplikasi ini menggunakan metode *Prototype*. *Prototyping* perangkat lunak (*software prototyping*) atau siklus hidup menggunakan *prototyping* (*life cycle using prototyping*) adalah salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat dari pada metode tradisional dan biayanya menjadi lebih rendah. Ciri khas dari metodologi ini adalah pengembangan sistem (*system developer*), *client* dan pengguna dapat melihat dan melakukan eksperimen dengan bagian dari sistem komputer dari sejak awal proses pengembangan.



Gambar 2. 1 Metode *Prototype*

Tahapan – tahapan dalam *Prototyping* adalah sebagai berikut:

1. Pengumpulan kebutuhan
Pelanggan dan pengembang bersama – sama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun *prototyping*
Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat *input* dan format *output*).
3. Evaluasi *prototyping*
Dilakukan oleh pengguna apakah *prototyping* yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak *prototyping* direvisi dengan mengulang langkah 1, 2, dan 3.
4. Mengkodekan sistem
Prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman *PHP Native* dan *Database MySQL*.
5. Menguji sistem
Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan.
6. Evaluasi sistem
Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan, jika tidak, ulangi langkah 4 dan 5.
7. Menggunakan sistem
Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

Terkait dengan pengujian program, pengujian yang akan digunakan adalah pengujian *black-box*. Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan merupakan alternatif dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang memungkinkan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*,
3. Kesalahan dalam struktur data atau akses *database* eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminal.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur - struktur kontrol, maka perhatian berfokus pada domain informasi.

Pengujian *black-box* ini terdapat beberapa proses. Proses-proses yang ada dalam pengujian ini diantaranya :

- a. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
- b. Pemilihan jenis *input* yang memungkinkan menghasilkan *output* dengan benar serta jenis *input* yang memungkinkan *output* salah pada perangkat lunak yang sedang diuji.
- c. Menentukan *output* untuk satu jenis *input*.
- d. Pengujian dilakukan dengan *input-input* yang telah benar-benar diseleksi.
- e. Melakukan pengujian
- f. Perbandingan *output* yang dihasilkan dengan *output* yang diharapkan.
- g. Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Pada tahap ini dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru BAAK.

2.2.3 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan proyek yang berisi data dan operasi yang diperlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.[8]

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus menjadi kelompok data dan fungsi.

Setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, komponen lainnya, dan dapat berinteraksi satu sama lain.




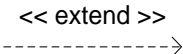

UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembang perangkat lunak berbasis objek (*Objek Oriented Programming*). Konsep pemodelan menggunakan *UML* sesungguhnya tidak ada batasan yang tegas diantara berbagai konsep dan konstruksi dalam *UML*, tetapi untuk menyederhanakannya, kita membagi sejumlah besar konsep dan dalam *UML* menjadi beberapa *view*. Suatu *view* sendiri pada dasarnya merupakan sejumlah konstruksi pemodelan *UML* yang mempresentasikan suatu aspek tertentu dari sistem atau perangkat lunak yang sedang kita kembangkan. Pada peringkat paling atas, *view – view* sesungguhnya dapat dibagi menjadi tiga area utama, yaitu klasifikasi struktural (*Structural classification*), perilaku dinamis (*dynamic behaviour*), serta pengolahan atau manajemen model (*model management*).[9]

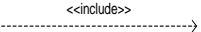
a. Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem tersebut skenario sedangkan pengguna disebut *actor*. *Actor* adalah sebuah peran yang biasa dimainkan oleh pengguna dalam interaksinya dengan sistem. Model *use case* adalah bagian dari model *requirement*. Definisi lain *use case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*.

Berdasarkan definisi diatas maka dapat disimpulkan bahwa *Use Case* adalah kontruks untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial yang terdiri dari sekumpulan skenario dan *actor*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta analis dan klien. Simbol *Use Case* dapat dilihat pada *Table 2.1* :

Table 2. 1 Simbol *Use Case Diagram*



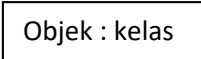
Simbol	Nama	Fungsi
	<i>Use Case</i>	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> . Biasanya dinyatakan dengan menggunakan kata kerja
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri
	<i>Association</i>	Komunikasi antar <i>actor</i> dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
	<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu
	<i>Generalization</i>	Hubungan generalisasi dari spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya






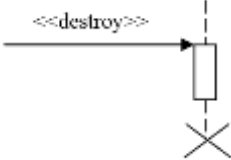
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini
---	----------------	---

b. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*[10]. Berikut adalah simbol-simbol yang ada pada *sequence diagram* dapat dilihat pada *Table 2.2* :

Table 2. 2 Simbol *Sequence Diagram*

Simbol	Nama	Fungsi
	<i>Aktor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat sendiri
	<i>Lifeline</i>	Menyatakan kehidupan suatu objek
	<i>Objek</i>	Menyatakan objek yang berinteraksi pesan

	<p><i>Waktu Aktif</i></p>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p><<create>></p> 	<p><i>Pesan Tipe Create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>1 : nama metode</p> 	<p><i>Pesan Tipe Call</i></p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>
	<p><i>Pesan Tipe Send</i></p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
	<p><i>Pesan Tipe Return</i></p>	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
<p><<destroy>></p> 	<p><i>Pesan Tipe Destroy</i></p>	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka <i>destroy</i></p>

2.2.4 Basis Data

Basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari *database* tersebut. Perangkat lunak yang digunakan untuk mengolah dan mengambil *query* basis data disebut sistem manajemen basis data.


Basis data atau juga disebut *database*, terdiri dari dua penggalan kata yaitu data dan *base*, yang artinya berbasiskan pada data, tetapi secara konseptual, *database* diartikan sebuah koleksi atau kumpulan data – data yang saling berhubungan (*relation*), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Selain itu, untuk mengelola dan memanggil *query* basis data agar dapat disajikan dalam berbagai bentuk yang diinginkan dibutuhkan perangkat lunak yang disebut Sistem Manajemen Basis Data atau juga disebut *Database Management System (DBMS)*. [11]



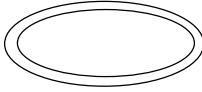
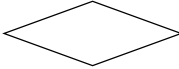
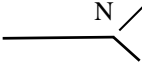
a. ERD

ERD adalah suatu rancangan atau bentuk hubungan suatu kegiatan di dalam sistem yang berkaitan langsung dan mempunyai fungsi di dalam proses tersebut. *ERD* adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship*. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya.

Berikut merupakan simbol-simbol dari *ERD*:

Table 2. 3 Simbol *ERD*

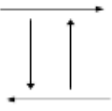

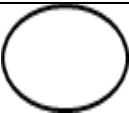
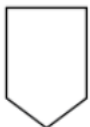

Simbol	Nama	Fungsi
	<i>Entitas /entity</i>	Entitas merupakan data inti yang akan disimpan; bakal <i>table</i> pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda










		dan belum merupakan nama tabel.
	<i>Atribut</i>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
	<i>Atribut primary key</i>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa <i>id</i> ; kunci primer dapat lebih dari satu kolom; asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
	<i>Atribut multinilai / multivalue</i>	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relasi</i>	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
	<i>Asosiasi / association</i>	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> yang menghubungkan entitas A dan entitas B.

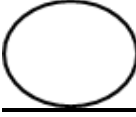


b. Flowchart Diagram

Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer. Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. *Flowchart* digambarkan dengan menggunakan simbol-simbol[12]. Berikut simbol-simbol pada *flowchart* dapat dilihat pada Tabel :

Table 2. 4 Simbol *Flowchart* Diagram

Simbol	Nama	Fungsi
	<i>Connecting Line</i>	Berfungsi untuk menghubungkan simbol yang satu dengan yang lainnya, menyatakan arus suatu proses
	<i>Communication Link</i>	Berfungsi untuk transmisi data dari satu lokasi ke lokasi lain
	<i>Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman yang sama
	<i>Offline Connector</i>	Digunakan untuk menyatakan sambungan dari proses yang satu ke proses berikutnya di halaman berbeda
	<i>Processing</i>	Digunakan untuk menunjukkan pengolahan yang akan dilakukan oleh komputer

	<i>Manual Operation</i>	Digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer
	<i>Decision</i>	Digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu
	<i>Predefined Process</i>	Digunakan untuk mempersiapkan penyimpanan yang sedang/akan digunakan dengan memberi harga awal
	<i>Terminal</i>	Digunakan untuk memulai atau mengakhiri program
	<i>Offline Storage</i>	Berfungsi untuk menunjukkan bahwa data akan disimpan ke media tertentu
	<i>Manual Input Symbol</i>	Digunakan untuk menginputkan data secara manual dengan <i>keyboard</i>
	<i>Input/Output</i>	Digunakan untuk menyatakan <i>input</i> dan <i>output</i> tanpa melihat jenisnya
	<i>Punched Card</i>	Digunakan untuk menyatakan masukan dan pengeluaran yang berasal dari <i>card</i>
	<i>Disk Storage</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari <i>disk</i>

	<i>Magnetic Tape</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari pita magnetis
	<i>Document</i>	Digunakan untuk menyatakan masukan dan keluaran yang berasal dari dokumen
	<i>Display</i>	Digunakan untuk menyatakan masukan dan keluaran melalui layer monitor

2.2.5 Peraturan Akademik (Permohonan Izin Mahasiswa)

Permohonan izin untuk tidak mengikuti perkuliahan adalah permohonan yang diproses di bagian admin jurusan yang berisikan keterangan bahwa mahasiswa tidak dapat mengikuti perkuliahan yang diadakan pada hari tersebut. Permohonan izin mahasiswa terdapat 3 jenis yaitu ketidakhadiran karena sakit, ketidakhadiran yang di izinkan dan ketidakhadiran yang tidak di izinkan. Dan juga terdapat peringatan tertulis digunakan jika mahasiswa/i melakukan ketidakhadiran yang tidak di izinkan.

a. Ketidakhadiran karena sakit terdapat di Peraturan Akademik Pasal 50

1. Ketidakhadiran karena sakit, harus disertakan surat keterangan dokter yang resmi dan sah.
2. PNC berhak memeriksa keabsahan surat keterangan dokter yang dijadikan alasan ketidakhadiran mahasiswa.
3. Surat keterangan dokter/pernyataan alasan absen yang ditandatangani oleh orang tua/wali harus diserahkan kepada Sub- bagian Administrasi Akademik dan Kemahasiswaan selambat – lambatnya ketika mengikuti kuliah kembali.
4. Ketidakhadiran karena sakit diberi kesempatan untuk mengulang matakuliah yang tidak diikuti dengan persetujuan dosen yang bersangkutan.

b. Ketidakhadiran karena sakit terdapat di Peraturan Akademik Pasal 50

1. Ketidakhadiran untuk mengikuti perkuliahan yang diizinkan dalam hal :
 - a. Urusan pribadi yang benar – benar penting dan mendadak (contoh : panggilan untuk menjadi saksi dimuka pengadilan, pelanggaran atau kecelakaan lalu lintas dan anggota keluarga yang mempunyai hubungan langsung mendapat kecelakaan berat). Izin untuk urusan pribadi diberikan sebanyak - banyaknya untuk 1(satu) hari.
 - b. Anggota keluarga yang mempunyai hubungan langsung (Ayah, Ibu, Kakak dan/atau Adik) meninggal dunia. Untuk kasus ini izin iberikan selama 2(dua) hari untuk daerah Kota Cilacap, dan 5(lima) hari untuk daerah di luar kota Cilacap.
2. Izin tidak hadir selama jam kuliah kurang dari satu hari harus didapat dari Ketua jurusan/Prodi.
3. Bila mahasiswa izin lebih dari 1(Satu) hari diwajibkan melampirkan surat keterangan pendukung lainnya seperti surat tugas/dispensasi dari PNC.
4. Bila mahasiswa karena suatu hal tidak dapat mengikuti perkuliahan karena alasan tersebut diatas, maka Ketua Prodi harus menerima pemberitahuan resmi dalam waktu selambat – lambatnya 3 hari kerja.

c. Ketidakhadiran yang tidak di izinkan terdapat di Peraturan Akademik Pasal 52.

1. Ketidakhadiran yang tidak di izinkan meliputi kasus :
 - a. Terlambat,
 - b. Bolos/Tanpa keterangan/alpha
2. Ketidakhadiran yang tidak di izinkan akan mengakibatkan diberikannya peringatan lisan atau peringatan tertulis.

d. Ketidakhadiran karena sakit terdapat di Peraturan Akademik Pasal 50

1. Peringatan tertulis dapat dikenakan kepada mahasiswa atas pelanggaran sebagai berikut :
 - a. Mengakibatkan terjadinya suatu kecelakaan karena pelanggaran terhadap intruksi keselamatan yang

- membahayakan kesehatan dan kehidupan bagi diri sendiri dan/atau orang lain,
- b. Tidak melaksanakan/mentaati tugas kompensasi pada waktu yang telah ditentukan,
 - c. Tidak jujur dalam memberikan alasan absen,
 - d. Tidak jujur dalam memberikan alasan/kerusakan alat/mesin/benda kerja,
 - e. Melakukan kegiatan politik di dalam kampus,
 - f. Hal – hal lain yang diterbitkan oleh unit kerja yang telah disetujui oleh Direktur.
2. Peringatan tertulis dapat diberikan oleh seluruh staf sesuai dengan lingkup tanggungjawabnya
 3. Nama mahasiswa dan jenis pelanggaran yang dikenakan ditulis dalam *Form* Peringatan serta ditandatangani oleh staf yang melaporkan, mahasiswa yang bersangkutan dan Ketua jurusan/Prodi
 4. PNC akan menyampaikan tembusan surat peringatan tertulis kepada orangtua/wali mahasiswa
 5. Pemberian surat peringatan tertulis yang didasarkan pada jumlah jam ketidakhadiran yang tidak diizinkan, misalnya terlambat dan bolos dengan ketentuan sebagai berikut :
 - 6.

Table 2. 5 Peraturan Akademik

Tidak hadir tanpa izin ≥ 10 jam atau	Surat Peringatan I	Masa berlaku 3 bulan, jika melakukan kesalahan yang sama akan mendapatkan SP 2
Tidak hadir tanpa izin ≥ 20 jam	Surat Peringatan II	Masa berlaku 6 bulan, jika melakukan kesalahan yang sama akan mendapatkan SP 3

Tidak hadir tanpa izin ≥ 30 jam	Surat Peringatan III	Masa berlaku 9 bulan, jika melakukan kesalahan yang sama akan mendapatkan DO
Tidak hadir tanpa izin ≥ 38 jam	Surat pemberhentian/dike luarkan dari Politeknik	Drop Out
Tidak hadir berturut – turut selama tiga hari tanpa izin	Surat pemberhentian/dike luarkan dari Politeknik	Drop Out

-HALAMAN INI SENGAJA DIKOSONGKAN-