

BAB II

LANDASAN TEORI

Dalam Bab ini akan diuraikan mengenai Prinsip dan Konsep Interaksi Manusia-Komputer (IMK), Model Interaksi Manusi-Komputer, Tujuan Interaksi Manusia-Komputer, Prinsip dan Konsep Antarmuka Grafis (GUI), Komponen Antarmuka Grafis (GUI), Sejarah dan pengenalan bahasa pemrograman Python, dan Pengenalan GUI di Python Secara Umum.

2.1 Prinsip dan Konsep Interaksi Manusia-Komputer (IMK)

Ketika komputer pertama kali diperkenalkan secara komersial pada tahun 50-an, mesin ini sangat sulit dipakai dan sangat tidak praktis. Hal demikian karena waktu itu komputer merupakan mesin yang sangat mahal dan besar, hanya dipakai kalangan tertentu, misalnya para ilmuwan atau ahli-ahli teknik.

Setelah komputer pribadi (PC) diperkenalkan pada tahun 70-an, maka berkembanglah penggunaan teknologi ini secara cepat dan mengagumkan ke berbagai penjuru kehidupan (pendidikan, perdagangan, pertahanan, perusahaan, dan sebagainya). Kemajuan-kemajuan teknologi tersebut akhirnya juga mempengaruhi rancangan sistem. Sistem rancangan dituntut harus bisa memenuhi kebutuhan pemakai, sistem harus mempunyai kecocokkan dengan kebutuhan pemakai atau suatu sistem yang dirancang harus berorientasi kepada pemakai. Pada awal tahun 70-an, juga mulai muncul isu teknik antarmuka pemakai (*user interface*) yang diketahui sebagai *Man-Machine Interaction (MMI)* atau Interaksi Manusia-Mesin.

Pada *Man-Machine Interaction* sudah diterapkan sistem yang "*user friendly*". Namun, sifat *user friendly* pada MMI ini diartikan secara terbatas. *User friendly* pada MMI hanya dikaitkan dengan aspek-aspek yang berhubungan dengan estetika atau keindahan tampilan pada layar saja. Sistem tersebut hanya menitik beratkan pada aspek rancangan antarmukanya saja, sedangkan faktor-faktor atau aspek-aspek yang berhubungan dengan pemakai baik secara organisasi atau individu belum diperhatikan [PRE94].

Para peneliti akademis mengatakan suatu rancangan sistem yang berorientasi kepada pemakai, yang memperhatikan kapabilitas dan kelemahan pemakai ataupun sistem (komputer) akan memberi kontribusi kepada interaksi manusia-komputer yang lebih baik. Maka pada pertengahan tahun 80-an diperkenalkanlah istilah *Human-Computer Interaction (HCI)* atau Interaksi Manusia-Komputer (IMK).

Interaksi manusia-komputer (*Human-Computer Interaction-HCI*) merupakan satu disiplin ilmu yang mengkaji tentang komunikasi atau interaksi di antara pengguna dengan sistem. Pada HCI ini cakupan atau fokus perhatiannya lebih luas, tidak hanya berfokus pada rancangan antarmuka saja, tetapi juga memperhatikan semua aspek yang berhubungan dengan interaksi antara manusia dan komputer. HCI ini kemudian berkembang sebagai disiplin ilmu tersendiri (yang merupakan bidang ilmu interdisipliner) yang membahas hubungan timbal balik antara manusia-komputer beserta efek-efek yang terjadi diantaranya.

Menurut Baecker dan Buxton [dalam PRE94] HCI ini didefinisikan sebagai *"set of processes, dialogues, and actions through-which a human user employs and interacts with computer"*. ACM-SGCHI [dalam PRE94] lebih jauh menuliskan definisi tentang HCI sebagai berikut:

" human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing system for human use and with the study of major phenomena surrounding them. "

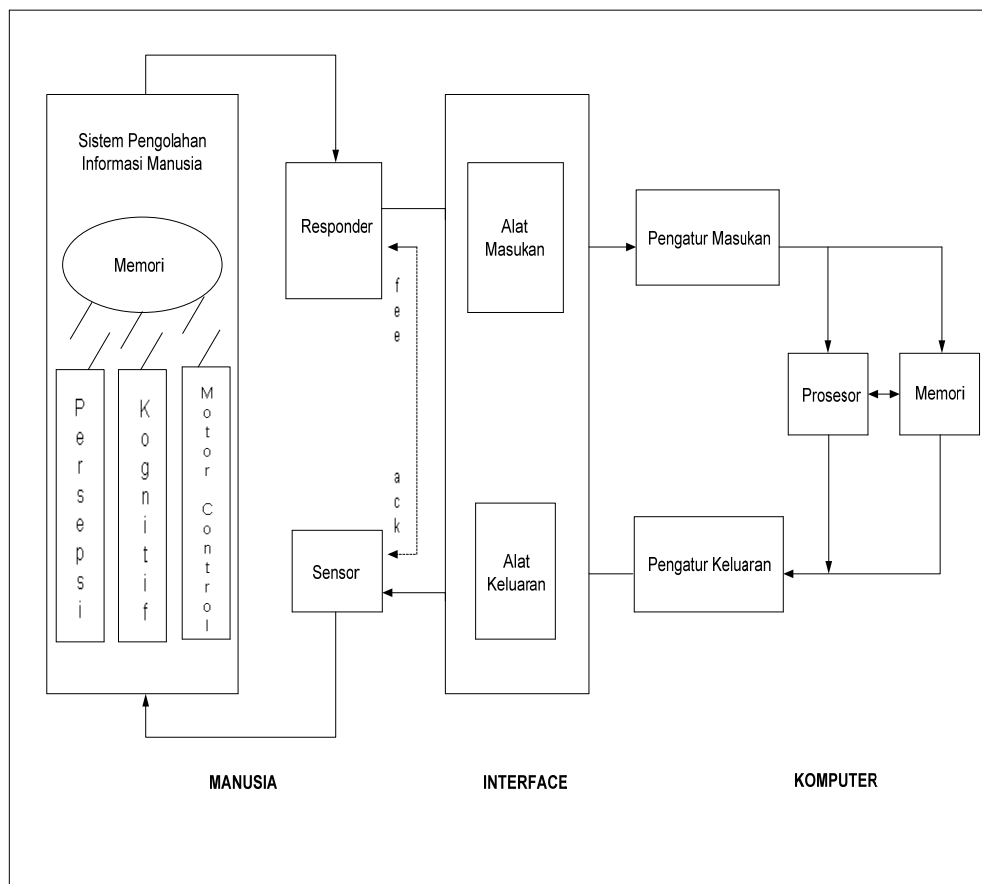
Dengan demikian terlihat jelas bahwa fokus perhatian HCI tidak hanya pada keindahan tampilannya saja atau hanya tertuju pada tampilan antarmukanya saja, tetapi juga memperhatikan aspek-aspek pemakai, implementasi sistem rancangannya dan fenomena lingkungannya, dan lainnya. Misalnya, rancangan sistem itu harus memperhatikan kenyamanan pemakai, kemudahan dalam pemakaian, mudah untuk dipelajari dan lain sebagainya.

2.2 Model Interaksi Manusia-Komputer

Dalam berinteraksi dengan komputer, para pemakai pertama kali akan berhadapan dengan perangkat keras komputer. Untuk sampai pada isi yang ingin disampaikan oleh perangkat lunak, pemakai dihadapkan terlebih dahulu dengan seperangkat alat seperti papan ketik (*keyboard*), monitor, *mouse*, *joystick*, dan lain-lain. Pemakai harus dapat mengoperasikan seperangkat alat tersebut. Selanjutnya, pemakai akan berhadapan dengan macam-macam tampilan menu, macam-macam perintah yang terdiri dari kata atau

kata-kata yang harus diketikkannya, misalnya *save*, *copy*, *delete*, atau macam-macam ikon. Peralatan, perintah, ikon dan lain-lain yang disebutkan di atas dikenal dengan nama *interface* (antarmuka). *Interface* ini merupakan lapisan pertama yang langsung bertatap muka dengan pemakai.

Sistem interaksi itu sendiri juga merupakan bagian dari sistem komputer yang dibuat, sehingga memungkinkan manusia berinteraksi dengan sistem komputer se-efektif mungkin guna memanfaatkan kemampuan pengolahan yang tersedia pada sistem komputer. Salah satu model antarmuka antara manusia dan komputer atau rangkaian kegiatan interaksi manusia-komputer secara sederhana dapat disimak pada gambar di bawah :



Gambar 2.1 Model Rangkaian Kegiatan Interaksi Manusia-Komputer [DOW92].

Dari Gambar 2.1 ini terlihat bahwa manusia memberi isyarat, atau masukan data kepada sistem pengolah informasi komputer, melalui alat masukan yang tersedia pada sistem komputer (misalnya *keyboard*). Berdasarkan masukan ini, melalui alat keluarannya (misalnya *monitor*), hasil-hasil pengolahan dari prosesor komputer disajikan kepada manusia. Melalui sensor manusia, seperti penglihatan (*mata*), pendengaran (*telinga*), dan peraba, sajian atau masukan-masukan itu kemudian dipantau/dimonitor untuk selanjutnya diteruskan ke sistem pengolah informasi manusia (*perceptual processing*, *intellectual/cognitive processing*, dan *motor control* yang selalu berinteraksi dengan *human memory*) untuk ditafsirkan.

Setelah penafsiran dilakukan dan keputusan diambil, maka diteruskanlah perintah ke alat responder manusia (*tangan*, *jari*, *suara*, dan lainnya) untuk melakukan tindak lanjutan yang pada umumnya diwujudkan berupa masukan kembali ke komputer.

Rangkaian pesan dan isyarat antara manusia dengan komputer membentuk suatu dialog interaktif, yakni serangkaian aksi dan reaksi yang saling berkaitan. Memperhatikan rangkaian kegiatan interaksi yang terjadi, jelaslah bahwa sifat dari suatu interaksi sangat ditentukan oleh dialog manusia-komputer dan teknologi dari alat masukan serta keluaran yang digunakan [DOW92].

Dengan faktor-faktor yang tercakup pada HCI tersebut, sekarang semakin menjadi jelas peranan HCI di dalam rancangan sistem. Ia akan mempertinggi kualitas interaksi antara sistem

komputer dan manusia. Untuk bisa tercapainya kualitas yang tinggi pada interaksi tersebut, maka secara sistematis perlu diterapkannya pengetahuan tentang tujuan manusia (*human goals*), kapabilitas dan keterbatasan manusia bersama-sama dengan pengetahuan tentang kapabilitas dan keterbatasan komputer pada sistem. Pengetahuan-pengetahuan yang berhubungan itu selanjutnya untuk mengetahui aspek-aspek fisik, sosial, organisasi dan lingkungan kerja pemakai. Dengan demikian, perancang sistem harus bisa membuat transisi dari 'apa yang dapat dilakukan kepada sistem (*functionality*)' menjadi 'bagaimana sistem itu harus dilakukan agar cocok dengan pemakai (*usability*)'.

2.3 Tujuan Interaksi Manusia-Komputer

Tujuan dari IMK adalah untuk menghasilkan sistem yang bermanfaat (*usable*) dan aman (*safe*), maksud dari bermanfaat (*usable*) disini yaitu manusia berinteraksi dengan sistem komputer se-efektif mungkin guna memanfaatkan kemampuan pengolahan yang tersedia pada sistem komputer. Kedua sistem tersebut bisa untuk mengembangkan dan meningkatkan keamanan (*safety*), utilitas (*utility*), ketergunaan (*usability*), efektifitas (*effectiveness*) dan efisiensinya (*eficiency*). Sistem yang dimaksud konteksnya tidak hanya pada perangkat keras dan perangkat lunak, tetapi juga mencakup lingkungan secara keseluruhan, baik itu lingkungan organisasi masyarakat kerja atau lingkungan keluarga. Sedangkan utilitas mengacu kepada fungsionalitas sistem atau sistem tersebut dapat meningkatkan efektifitas dan efesiensi kerjanya. Ketergunaan (*usability*) disini dimaksudkan bahwa sstem yang dibuat

tersebut mudah digunakan dan mudah dipelajari baik secara individu ataupun kelompok.

Pendapat Preece, J. di atas didasarkan pada pemikiran yang menyatakan bahwa kepentingan pemakai sistem harus didahulukan, pemakai tidak bisa diubah secara radikal terhadap sistem yang telah ada, sistem yang dirancang harus cocok dengan kebutuhan-kebutuhan pemakai.

Selanjutnya, dalam berinteraksi dengan komputer, para pemakai pertama kali akan berhadapan dengan perangkat keras komputer. Untuk sampai pada isi yang ingin disampaikan oleh perangkat lunak, pemakai dihadapkan terlebih dahulu dengan seperangkat alat seperti papan ketik (*keyboard*), monitor, *mouse*, *joystick*, dan lain-lain. Pemakai harus dapat mengoperasikan seperangkat alat tersebut. Selanjutnya, pemakai akan berhadapan dengan macam-macam tampilan menu, macam-macam perintah yang terdiri dari kata atau kata-kata yang harus diketikkannya, misalnya *save*, *copy*, *delete*, atau macam-macam ikon. Peralatan, perintah, ikon dan lain-lain yang disebutkan di atas dikenal dengan nama *interface* (antarmuka). *Interface* ini merupakan lapisan pertama yang langsung bertatap muka dengan pemakai.

2.4 Prinsip dan Konsep Antarmuka Grafis (GUI)

Pada beberapa tahun terakhir ini, istilah antarmuka grafis (graphical user interface, GUI) semakin banyak terdengar. Istilah ini merujuk pada suatu program antarmuka berbasis grafis, dimana perintah-perintah tidak lagi diketikan lewat papan ketik, tetapi

dengan cara melakukan interaksi langsung terhadap apa yang terlihat pada layar, yang sebenarnya merupakan suatu abstraksi dari suatu perintah kepada komputer agar komputer mengerjakan apa yang kita inginkan.

Pemakaian antarmuka grafis menjadi semakin meningkat, setelah keberhasilan cara ini dipakai pada Microsoft Windows maupun pada Apple Macintosh. Pada kedua sistem ini, layar komputer menyajikan objek-objek grafis untuk menggambarkan sesuatu yang lebih nyata, misalnya kertas, arsip, lemari arsip, keranjang sampah, dan lain-lain. Dengan menggunakan mouse, pengguna sistem mengklik suatu gambar pada layar, yang selanjutnya disebut ikon (*icon*), dan sistem akan mengerjakan semua perintah yang "disembunyikan" didalam ikon tersebut. Pengguna sistem tidak perlu mengetahui bagaimana perintah-perintah tersebut dikerjakan, tetapi hasilnya bisa segera dilihat pada layar.

Dengan semakin pesatnya perkembangan teknologi grafika komputer, khususnya dalam hal pemanfaatan raster display, dan juga atas keberhasilan dari Microsoft Windows maupun Apple Macintosh, antarmuka grafis semakin menarik perhatian berbagai pihak. Salah satu yang paling menarik perhatiannya, yaitu perintah-perintah untuk mengerjakan suatu aktifitas tidak perlu diketikkan lewat papan ketik, tetapi diganti dengan hanya mengklik suatu ikon yang terlihat dilayar. Dengan variasi ikon yang sangat menarik, akan membuat program menjadi lebih hidup.

2.5 Komponen Antarmuka Grafis (GUI)

Secara umum, komponen-komponen dasar dari antarmuka grafis adalah tombol, penggeser atau penggulung layar, dan ikon, serta piranti masukan yang biasanya menggunakan mouse.

Tombol biasanya digunakan untuk menampilkan pilihan-pilihan yang tersedia, menunjukkan perubahan nilai parameter atau untuk menunjukan pilihan-pilihan yang berstatus aktif atau pasif. Beberapa macam tombol yang sering digunakan antara lain adalah tombol tekan (*push button*), *check button*, dan *radio button*.

Penggeser layar adalah komponen kedua dari piranti antarmuka yang hampir selalu ada. Seperti diketahui, informasi yang akan ditampilkan di layar seringkali melebihi ukuran layar yang sebenarnya, sehingga tidak semua informasi bisa tampil secara utuh. Kegunaan dari penggeser adalah untuk menggulung layar agar informasi berikutnya, atau sebelumnya, bisa ditampilkan pada layar. Sesuai dengan ukuran layar, maka penggeseran layar bisa dilakukan ke arah vertikal dan ke arah horizontal.

2.6 Sejarah dan Pengenalan Bahasa Pemrograman Python

Pada awalnya kita akan beranggapan bahwa penamaan bahasa pemrograman ini didasarkan pada binatang melata, anggapan tersebut salah. Penamaan bahasa pemrograman ini berdasarkan pembuatnya ketika Pembuat python ini menonton acara komedi di televisi di BBC yang bernama *Monty Python's Flying Circus*. Pembuat bahasa pemrograman ini adalah Guido van Rossum dari CWI (*Centrum voor Wiskunde en Informatica*) Amsterdam, Belanda

pada tahun 1990. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba.

Python merupakan bahasa yang sangat sederhana, bahasa yang sangat mudah dipelajari sehingga dapat membuat biaya perawatan sebuah program menjadi murah. Semua source Interpreter Python dan tambahan library – library standarnya tersedia untuk digunakan oleh semua orang dan dapat diperoleh dengan gratis.

Bahasa pemrograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan menggunakan python sebagai pembuat perangkat lunak komersial.

Sejak muncul tahun 1991 di domain public, bahasa pemrograman ini berkembang dengan dukungan komunitas pengguna dan pengembangnya, seperti *Python Software Activity*, internet newsgroup *comp.lang.python*, dan organisasi informal lainnya.

Python merupakan *interpreter* bahasa pemrograman tingkat tinggi berbasis objek dengan semantik yang dinamis, dimana bersifat freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, antarmuka pengguna grafis (GUI), dan basis datanya. Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada

kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya.

2.6.1 Fitur-fitur pada Bahasa Pemrograman Python

Bahasa pemrograman python memiliki beberapa fitur yang terdapat di dalamnya diantaranya :

- Memiliki kepustakaan yang luas
Dalam distribusi Python telah disediakan modul-modul 'siap pakai' untuk berbagai keperluan.
- Memiliki tata bahasa yang mudah dipelajari.
- Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
- Berorientasi objek.
- Memiliki sistem pengelolaan memori otomatis (garbage collection, seperti java)
- Modular.
Mudah dikembangkan dengan menciptakan modul-modul baru, modul-modul tersebut dapat dibangun dengan bahasa Python maupun C/C++.

2.6.2 Filosofi pada Bahasa Pemrograman Python

Bahasa pemrograman ini mempunyai filosofi sendiri, antara lain adalah:

➤ *Coherence.*

Bahasa pemrograman skrip tidaklah menjadi sulit untuk dibaca, ditulis, dan dimaintain. *Power (kekuatan)*. Bahasa pemrograman ekstensi tidaklah mempunyai fungsi yang terbatas.

➤ *Scope (jangkauan).*

Bahasa pemrograman dinamis dapat digunakan untuk berbagai macam tugas. Tidak ada alasan bahwa bahasa pemrograman tidak dapat menyediakan tanggapan yang cepat selama masa pembuatan sistem aplikasi dan juga mempunyai kelebihan yang membuatnya berguna untuk membuat lebih daripada sistem aplikasi tingkat tinggi.

➤ *Object.*

Pemrograman berorientasi objek dapat menjadi paradigma yang berguna, karena pemrograman berorientasi objek mempunyai kekuatan dalam strukturnya dan penggunaan kembali kode yang telah ada. Bahasa pemrograman berorientasi objek tidak perlu untuk menjadi kompleks atau sulit untuk digunakan.

➤ *Integration* (integrasi).

Tidak ada bahasa pemrograman yang menjadi pulau. Penyediaan dari kekuatan bahasa pemrograman dinamis, dan antarmuka yang terdefinisi dengan baik untuk bahasa pemrograman lainnya, python membantu sistem hibrid yang secara simultan pengungkitan kecepatan perubahan dari python, dan efisiensi dari C.

Bahasa pemrograman python cenderung untuk menyerupai bahasa pemrograman tradisional seperti C dan Pascal, daripada bahasa pemrograman skrip seperti Perl dan Tcl. Pada kenyataannya, Python adalah suatu dari bahasa pemrograman skrip dalam samaran dari bahasa pemrograman tradisional. Ada pendapat lain yang menyatakan bahwa Python merupakan jembatan antara bahasa skrip dan C.

2.6.3 Kelebihan dan Keuntungan menggunakan Bahasa Pemrograman Python

Bahasa pemrograman ini mempunyai kelebihan dari beberapa bahasa pemrograman skrip dan bahasa pemrograman C.

Tabel 2.1 kelebihan dan keuntungan Python

NO	KELEBIHAN	KEUNTUNGAN
1	Tidak ada tahapan kompilasi dan penyambungan (link).	Kecepatan perubahan pada masa pembuatan sistem aplikasi.
2.	Tidak ada deklarasi tipe.	Program menjadi lebih sederhana, singkat, dan lebih fleksible.
3.	Manajemen memori otomatis.	Kumpulan sampah memori sehingga dapat menghindari pencatatan kode.

NO	KELEBIHAN	KEUNTUNGAN
4.	Tipe data dan operasi tingkat tinggi.	Kecepatan pembuatan sistem aplikasi menggunakan tipe objek yang telah ada.
5.	Pemrograman berorientasi objek.	Penstrukturan kode dan penggunaan kembali, integrasi C++.
6.	Pelekatan dan perluasan dalam C.	Integrasi C, sistem bahasa campuran
7.	Kelas, Modul, Eksepsi.	Dukungan pemrograman skala besar secara modular.
8.	Pemuatan dinamis modul C.	Ekstensi yang sederhana, serta berkas biner yang lebih kecil.
9.	Pemuatan kembali secara dinamis modul Python.	Memodifikasi aplikasi tanpa menghentikannya.
10.	Model objek universal kelas satu.	Fewer restriction dan special-case rules.
11.	Konstruksi pada saat aplikasi berjalan.	Handle unforeseen need, end-user coding.
12.	Interaktif, dinamis alamiah.	Incremental development and testing.
13.	Akses hingga informasi interpreter.	Metaprogramming, introspective object.
14.	Portabilitas secara luas.	Pemrograman antar platform tanpa ports.
15.	Kompilasi untuk portabel kode byte.	Kecepatan eksekusi, melindungi kode sumber.
16.	Antarmuka terpasang untuk pelayanan keluar.	Perangkat bantu sistem, GUI, persistence, database, dan lain sebagainya.

(Sumber : Anonim, 2006)

Penggunaan python pada lingkungan yang berbagai macam ragam, sulit untuk menjawab secara resmi terhadap pertanyaan di atas. Pada bagian bawah ini merupakan rangkuman dari beberapa bidang yang menggunakan python pada saat ini:

- *Perangkat bantu shell.* Tugas-tugas sistem administrator, program baris perintah.
- *Kerja bahasa ekstensi.* Antarmuka untuk pustaka C/C++, kustomisasi
- *Pembuatan prototipe secara cepat/pembuatan sistem aplikasi.* Prototipe yang dapat dibuang atau sesuai dengan permintaan.
- *Modul berdasarkan bahasa pemrograman.* Pengganti dari penulisan parser khusus.
- *Antarmuka pengguna grafis.* Penggunaan GUI API sederhana dan canggih.
- *Pengaksesan basisdata.* Penyimpanan objek tetap, antarmuka sistem SQL.
- *Pemrograman terdistribusi.* Penggunaan API mekanisme client/server terintegrasi.
- *Skrip internet.* Skrip CGI, antarmuka HTTP, Aplet WWW, dan lainnya.

Pada pengertian umum, python memberikan pelayanan terhadap jangkauan bidang – bidang, diantaranya:

- *Bahasa pemrograman dinamis*

Untuk situasi dimana salah satu tahap link atau kompilasi tidak memungkinkan (modifikasi di tempat klien), atau tidak nyaman (pembuatan prototipe, pembuatan aplikasi secara cepat, perangkat bantu shell).

- *Bahasa pemrograman yang powerful.*

Untuk situasi dimana kompleksitas dari bahasa pemrograman yang besar is (prototipe, pengkodean pada pemakai akhir).

- *Generalized language tool.*

Untuk situasi dimana dibutuhkan untuk menanamkan dan implementasi bahasa pemrograman kecil lainnya (antarmuka sistem yang dapat di program, perangkat bantu konfigurasi).

2.6.4 Kekurangan Pada bahasa Pemrograman Python

Beberapa penugasan terdapat diluar dari jangkauan python, seperti bahasa pemrograman dinamis lainnya, python tidak secepat atau efisien sebagai statis, tidak seperti bahasa pemrograman kompilasi seperti bahasa C. Pada beberapa bidang hal ini tidak menjadi pertimbangan utama, tetapi dalam bidang lainnya, efisiensi menjadi prioritas tertinggi.

Disebabkan python merupakan interpreter, python bukan merupakan perangkat bantu terbaik untuk pengantar komponen performa kritis. Sebaliknya, penggunaan operasi komputasi intensif dapat diimplementasikan sebagai ekstensi kompilasi bagi python, dan dikodekan dalam bahasa pemrograman tingkat rendah seperti bahasa C. Python tidak dapat digunakan sebagai dasar bahasa pemrograman implementasi untuk beberapa komponen,

tetapi dapat bekerja dengan baik sebagai bagian depan skrip antarmuka.

2.7 Pengenalan GUI di Python

Python merupakan bahasa pemrograman yang bersifat *open source* (sumber terbuka) yang berarti bahwa dibebaskan untuk setiap orang menggunakan dan sumber tersebut tersedia baik untuk dilihat ataupun untuk dimodifikasi. Atau setiap orang dapat memberikan masukan dalam hal memperbaiki sumber tersebut untuk kemajuan project secara keseluruhan. Dalam membuat suatu program berbasis grafis, di dalam python memerlukan pustaka GUI yang mendukung untuk pembuatan program grafis tersebut.

Ada lebih dari satu pustaka yang mendukung dalam pembuatan program berbasis grafis pada python, namun dalam tugas akhir ini penulis hanya akan membahas salah satu pustaka GUI yang mendukung pada python yaitu pustaka “*Tkinter*”. Salah satu pustaka yang terdapat langsung di dalam Python yaitu Tkinter. Tkinter adalah sebuah pustaka grafis (GUI widget) yang telah di set untuk Python sendiri. Dalam membangun sebuah aplikasi antarmuka grafis pada Python, pustaka Tkinter sangat mudah dan cepat dalam prosesnya, dan dapat secara langsung ditambahkan kerangka komponen yang lebih besar di Python itu sendiri.