

**PERKEMBANGAN BAHASA PEMROGRAMAN KOMPUTER DI
AMERIKA SERIKAT TAHUN 1955 - 1995**

SKRIPSI



**UNIVERSITAS ISLAM NEGERI
KIAI HAJI ACHMAD SIDDIQ
J E M B E R**

Oleh:

Fitria Dewi Rochmawati

NIM U20184025

UIN

UNIVERSITAS ISLAM NEGERI

**UNIVERSITAS ISLAM NEGERI
KIAI HAJI ACHMAD SIDDIQ JEMBER**

FAKULTAS USHULUDDIN, ADAB DAN HUMANIORA

DESEMBER, 2022

**PERKEBANGAN BAHASA PEMROGRAMAN KOMPUTER DI
AMERIKA SERIKAT TAHUN 1955 - 1995**

SKRIPSI

Diajukan kepada Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember
untuk memenuhi salah satu persyaratan memperoleh

gelar Sarjana Humaniora (S.Hum)

Fakultas Ushuluddin, Adab dan Humaniora

Program Studi Sejarah Peradaban Islam



Oleh:

Fitria Dewi Rochmawati

NIM U20184025

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ

UNIVERSITAS ISLAM NEGERI

KIAI HAJI ACHMAD SIDDIQ JEMBER

FAKULTAS USHULUDDIN, ADAB DAN HUMANIORA

DESEMBER, 2022

**PERKEMBANGAN BAHASA PEMROGRAMAN KOMPUTER DI
AMERIKA SERIKAT TAHUN 1955 – 1995**

SKRIPSI

Diajukan kepada Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember
untuk memenuhi salah satu persyaratan memperoleh

gelar Sarjana Humaniora (S.Hum)

Fakultas Ushuluddin, Adab dan Humaniora

Program Studi Sejarah Peradaban Islam

Oleh:

Fitria Dewi Rochmawati

NIM U20184025

Disetujui Pembimbing



Mawardi Purbo Sanjoyo, M.A

NIP 1990052820180111001

**PERKEMBANGAN BAHASA PEMROGRAMAN KOMPUTER
DI AMERIKA SERIKAT TAHUN 1955 - 1995**

SKRIPSI

Telah diuji dan diterima untuk memenuhi salah satu
persyaratan memperoleh gelar Sarjana Humaniora (S.Hum)
Fakultas Ushuluddin, Adab dan Humaniora
Program Studi Sejarah Peradaban Islam

Hari: Senin
Tanggal: 9 Januari 2023

Tim Penguji

Ketua

Dr. Win Usuluddin, M.Hum.
NIP. 197001182008011012

Sekretaris

Muhammad Faiz, M.A.
NIP. 198510312019031003

Anggota :

1. Dr. Akhiyat, S.Ag., M.Pd.

()

2. Mawardi Purbo Sanjoyo, M.A.

()

Menyetujui
Dekan Fakultas Ushuluddin, Adab dan Humaniora

Prof. Dr. M. Khuisja Amal, S.Ag., M.Si.
NIP. 19721208 199803 001

MOTTO

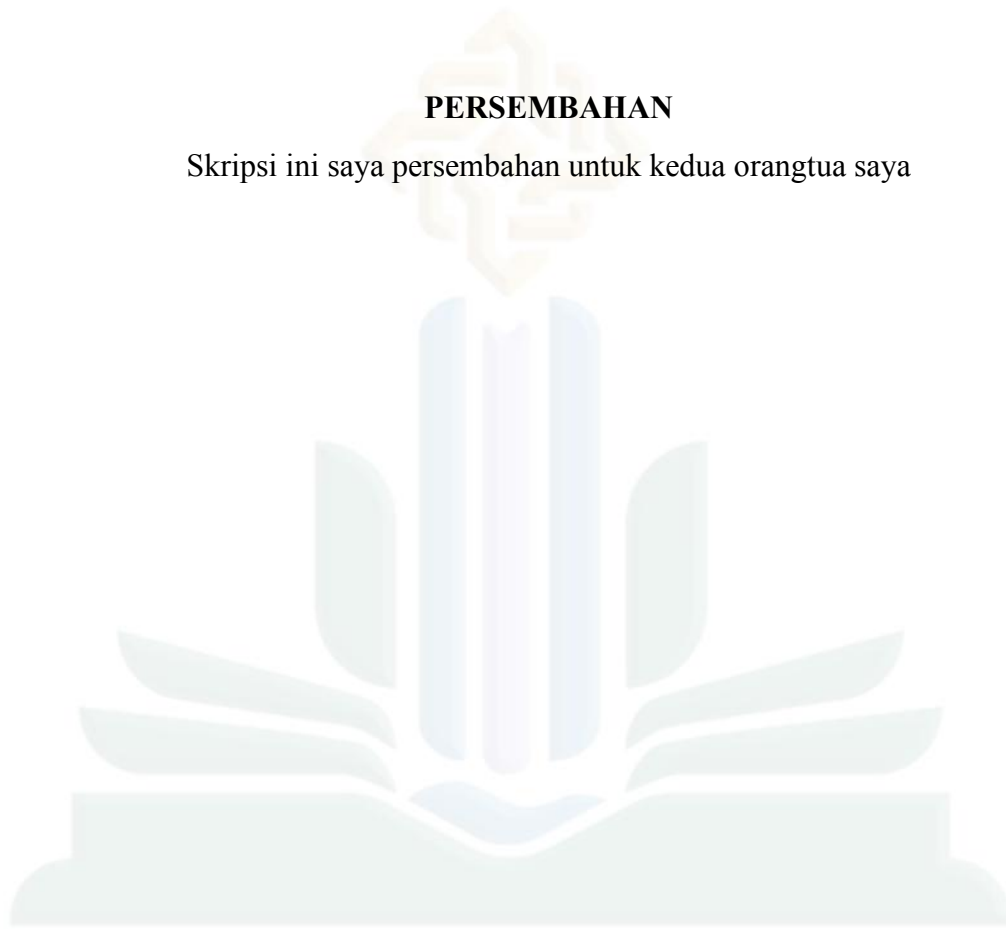
نَّ فِي خَلْقِ السَّمَاوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ وَالْفُلْكِ الَّتِي تَجْرِي فِي الْبَحْرِ بِمَا يَنْفَعُ النَّاسَ وَمَا أَنْزَلَ اللَّهُ مِنَ السَّمَاءِ مِنْ مَاءٍ فَأَحْيَا بِهِ الْأَرْضَ بَعْدَ مَوْتِهَا وَبَثَّ فِيهَا مِنْ كُلِّ دَابَّةٍ وَتَصْرِيفِ الرِّيَّاحِ وَالسَّحَابِ الْمُسَخَّرِ بَيْنَ السَّمَاءِ وَالْأَرْضِ لَآيَاتٍ لِقَوْمٍ يَعْقِلُونَ ﴿١٦٤﴾

Sesungguhnya pada penciptaan langit dan bumi, pergantian malam dan siang, kapal yang berlayar di laut dengan (muatan) yang bermanfaat bagi manusia, apa yang diturunkan Allah dari langit berupa air, lalu dengan itu dihidupkan-Nya bumi setelah mati (kering), dan Dia tebarkan di dalamnya bermacam-macam binatang, dan perkisaran angin dan awan yang dikendalikan antara langit dan bumi, (semua itu) sungguh, merupakan tanda-tanda (kebesaran Allah) bagi orang-orang yang mengerti. (Q.S Al- Baqarah: 164).¹

¹ Syekh Dr. Usman Thaha Hafizhahullah, *Al-Quran Terjemahan*, (Banten: Forum Pelayan Al-Quran, 2016), 176.

PERSEMBAHAN

Skripsi ini saya persembahkan untuk kedua orangtua saya



UIN

UNIVERSITAS ISLAM NEGERI

**KH ACHMAD SIDDIQ
JEMBER**

KATA PENGANTAR

Puji syukur Alhamdulillah penulis ucapkan kepada Allah SWT, karena dengan limpahan rahmat, taufik, hidayah dan inayah-Nyalah, perencanaan, pelaksanaan dan penyelesaian penulisan skripsi ini dapat terselesaikan dengan lancar. Sebagai tanda rasa syukur penulis, semua pengalaman selama proses penulisan skripsi akan penulis jadikan sebagai refleksi atas diri penulis untuk kemudian akan penulis implementasikan dalam bentuk sikap dan perilaku konstruktif dan produktif untuk kebaikan dan perbaikan semua warga bangsa. Terselesainya penulisan skripsi ini, penulis sadari karena bantuan dan peran berbagai pihak.

Oleh karena itu, penulis menyadari dan menyampaikan terima kasih yang sedalam-dalamnya kepada:

1. Rektor Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember Prof. Dr. H. Babun Suharto, SE., MM atas kesempatan dan fasilitas yang diberikan kepada penulis untuk mengikuti dan menyelesaikan pendidikan Program Sarjana.
2. Dekan Fakultas Ushuluddin, Adab dan Humaniora Prof. Dr. M. Khusna Amal, S.Ag., M.Si dan seluruh jajaran Dekanat yang lain atas kesempatan yang diberikan kepada penulis untuk menjadi mahasiswa Program Studi Sejarah Peradaban Islam pada Program Sarjana Fakultas Ushuluddin, Adab dan Humaniora Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember.
3. Ketua Jurusan Studi Islam Dr. Win Ushuluddin, M.Hum atas saran, motivasi serta diskusi-diskusi yang menarik dan membangun selama proses perkuliahan.

4. Koordinator Program Studi Sejarah Peradaban Islam Dr. Akhiyat, S.Ag., M.Pd atas bimbingan, motivasi serta diskusi-diskusi yang menarik dan membangun selama proses perkuliahan.
5. Dosen Pembimbing Mawardi Purbo Sanjoyo, M.A yang selalu memberikan motivasi dan meyakinkan penulis bisa menyelesaikan penulisan skripsi ini. Tanpa bimbingan, saran, bantuan, dan motivasi beliau penulisan skripsi ini tidak akan selesai.
6. Seluruh dosen di Program Studi Peradaban Islam Fakultas Ushuluddin, Adab dan Humaniora Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember yang dengan sukarela mentransfer, membagi teori-teori dan ilmu-ilmu serta pengalamannya selama proses perkuliahan.
7. Seluruh pegawai dan karyawan di lingkungan Fakultas Ushuluddin, Adab dan Humaniora Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember atas informasi-informasi yang diberikan yang sangat membantu penulis mulai dari awal kuliah sampai bisa menyelesaikan penulisan skripsi ini.
8. Teman-teman yang telah membantu dan memberi dukungan untuk menyelesaikan penulisan skripsi ini.

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

Akhirnya semoga segala amal baik yang telah dilakukan mendapat balasan yang sebaik mungkin dari Allah SWT. Atas segala kekurangan serta kekhilafan yang ada, sepuh hati penulis minta maaf yang sebesar-besarnya.

Jember, 12 Desember 2022

Penulis



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

ABSTRAK

Fitria Dewi Rochamawati. 2022, Perkembangan Bahasa Pemrograman Komputer Di Amerika Serikat Tahun 1955 – 1995

Bahasa pemrograman komputer adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Tujuan dari pemrograman adalah untuk membuat suatu program yang dapat melakukan suatu perhitungan atau pekerjaan sesuai dengan keinginan yang membuat pemrogram. Pemrograman dapat diartikan sebagai proses menulis, menguji dan memperbaiki (debug), dan memelihara kode yang membangun sebuah program dimana kode ini ditulis dalam berbagai bahasa pemrograman. Bahasa pemrograman komputer pertama yaitu bahasa mesin yang terdiri dari bilangan angka-angka, seiring berjalannya waktu bahasa pemrograman dikembangkan agar mempermudah untuk membuat suatu program.

Fokus dalam penelitian ini ada dua yaitu: (1) Bagaimana gambaran bahasa pemrograman yang muncul pada tahun 1955-1995? (2) Bagaimana dampak dari perkembangan bahasa pemrograman pada tahun 1955-1995?.

penelitian ini juga bertujuan untuk mengetahui dampak dari perkembangan bahasa pemrograman dalam perkembangan teknologi masa kini. Penelitian ini menggunakan pendekatan historis perspektif diakronik. Sumber datanya yaitu buku karya Jean E. Sammet yang berjudul *Programming Languages: History and Fundamentals*, *History of Programming Languages* karya Richard G. Gibson, Thomas J. Bergin, *Bahasa Pemrograman* karya Zinsari, *History of Programming Languages* karya Richard L. Wexelblat, dan *Concepts of Programming Languages* karya Robert W. Sebesta.

Berdasarkan hasil penelitian ini dapat diketahui bahwa terciptanya bahasa pemrograman beriringan dengan sejarah mesin dan komputer. Grace merancang sebuah *compiler*, yaitu perangkat yang menerjemahkan bahasa pemrograman yang dimengerti manusia, menjadi bahasa yang dimengerti komputer. Bahasa pemrograman tingkat rendah adalah bahasa pemrograman yang menggunakan set instruksi khusus dari sebuah prosesor yang sudah ditentukan oleh pembuatnya. Bahasa pemrograman tingkat tinggi pertama adalah bahasa pemrograman *COBOL* merupakan bahasa pemrograman yang didedikasi untuk keperluan bisnis, keuangan, dan sistem administrasi. Untuk keperluan sains, bahasa pemrograman yang diciptakan adalah *Fortran* yang dikembangkan oleh *IBM*. Bahasa pemrograman *Fortran* telah mengandung *IF*, *DO*, dan *GOTO* yang merupakan loncatan besar dalam perkembangan bahasa pemrograman. Pada saat ini bahasa pemrograman yang sangat bereperan penting dalam kemajuan teknologi adalah bahasa pemrograman *Python*.

Kata Kunci : Bahasa pemrograman Komputer, manfaat perkembangan bahasa pemrograman.

DAFTAR ISI

| | |
|---|-----|
| HALAMAN JUDUL DEPAN | |
| HALAMAN JUDUL DALAM | i |
| HALAMAN PERSETUJUAN PEMBIMBING | ii |
| HALAMAN PENGESAHAN TIM PENGUJI | iii |
| MOTTO | iv |
| PERSEMBAHAN | v |
| KATA PENGANTAR | vi |
| ABSTRAK | ix |
| DAFTAR ISI | x |
| DAFTAR GAMBAR | xii |
| BAB 1 PENDAHULUAN | |
| A. Konteks Penelitian | 1 |
| B. Fokus Penelitian | 4 |
| C. Ruang Lingkup Penelitian | 5 |
| D. Tujuan Penelitian | 5 |
| E. Manfaat Penelitian | 6 |
| F. Studi Terdahulu | 6 |
| G. Kerangka Konseptual | 13 |
| H. Metode Penelitian | 15 |
| I. Sistematika Pembahasan | 17 |
| BAB II SEJARAH BAHASA PEMROGRAMAN SEBELUM 1955 | |
| A. Sejarah munculnya bahasa pemrograman | 18 |
| B. Bahasa pemrograman tingkat rendah | 21 |
| BAB III PERKEMBANGAN BAHASA PEMROGRAMAN TAHUN 1955-1995 | |
| A. Bahasa pemrograman tingkat tinggi | 27 |
| B. Konferensi HOPL | 46 |
| BAB IV DAMPAK PERKEMBANGAN BAHASA PEMROGRAMAN | |
| A. Dampak perkembangan bahasa pemrograman dalam kemajuan teknologi masakini | 56 |
| BAB V PENUTUP | |
| A. Kesimpulan | 63 |
| B. Saran | 65 |

DAFTAR PUSTAKA..... 66

Lampiran - lampiran



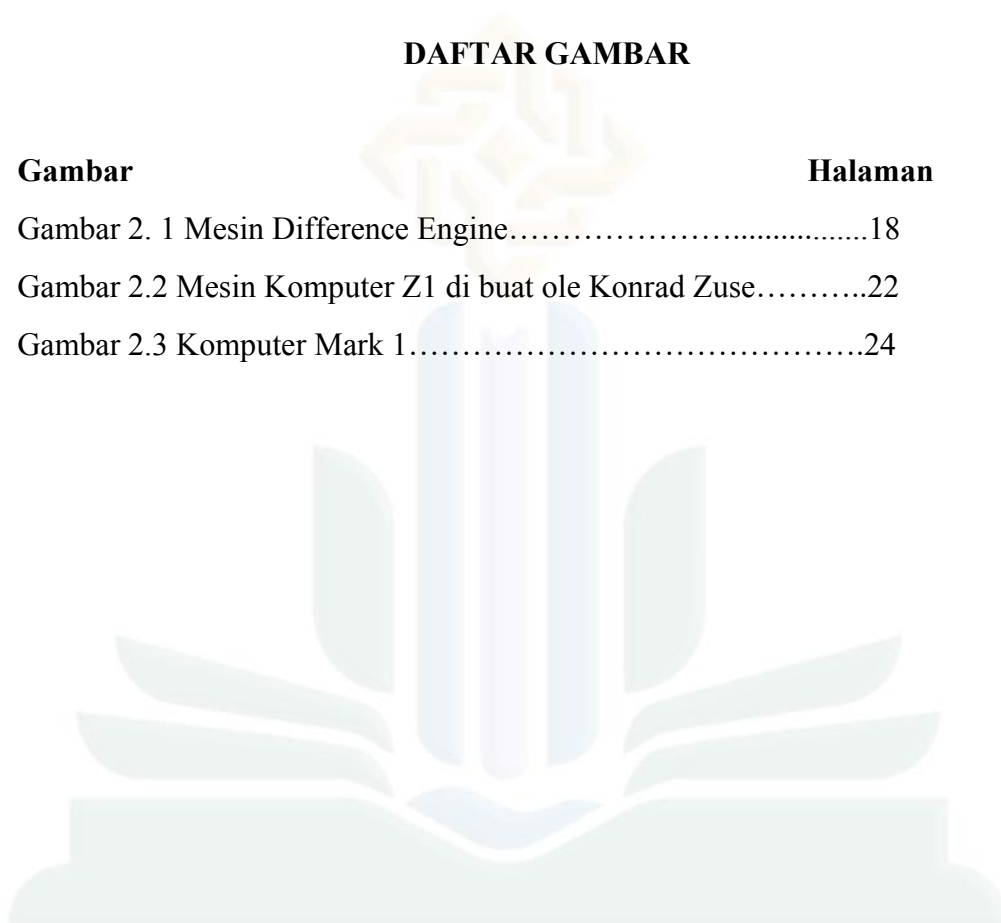
UIN

UNIVERSITAS ISLAM NEGERI

**KH ACHMAD SIDDIQ
JEMBER**

DAFTAR GAMBAR

| Gambar | Halaman |
|---|----------------|
| Gambar 2. 1 Mesin Difference Engine..... | 18 |
| Gambar 2.2 Mesin Komputer Z1 di buat ole Konrad Zuse..... | 22 |
| Gambar 2.3 Komputer Mark 1..... | 24 |



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

BAB I

PENDAHULUAN

A. Konteks Penelitian

Pemrograman komputer adalah dasar dari era digital yang dijalani saat ini. Perangkat digital apa pun yang digunakan dalam kehidupan sehari-hari, bergantung pada algoritma yang dikembangkan melalui bahasa pemrograman komputer. Setiap kali kita menyukai posting di media sosial, mengirim email, menyetel Jadwal Sholat di telepon genggam, atau saat mencari informasi sebuah situs sejarah di internet. bahasa pemrogramanlah yang bekerja di belakang layar. Bahasa pemrograman komputer adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Orang yang membuat program komputer disebut *Programmer*.¹

Tujuan dari pemrograman adalah untuk membuat suatu program yang dapat melakukan suatu perhitungan atau pekerjaan sesuai dengan keinginan yang membuat program. Untuk dapat melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan di banyak kasus, pengetahuan-pengetahuan lain seperti matematika.²

Pemrograman dapat diartikan sebagai proses menulis, menguji dan memperbaiki (debug), dan memelihara kode yang membangun sebuah program dimana kode tersebut ditulis dalam berbagai bahasa pemrograman. Sebelum

¹ Thomas. J,” A History of the History of Programming Languages”, *Jurnal Commutation of the ACM*, Vol 50 No.5, (Mei 2007): 72, <https://dl.acm.org/doi/10.1145/1230819.1230841>.

²Asnurul Isroqmi.“Kemampuan Mahasiswa Memahami Logika Pemrograman Komputer Melalui Algoritma”, *Jurnal Pendidikan Matematika*, Vol. 2 No. 2, (November 2017):13, <http://ejournal.unitaspalembang.ac.id/index.php/nabla/article/view/43>

membangun sebuah program komputer, perlu dibuat langkah-langkah penyelesaian, yaitu algoritma.³ Algoritma adalah sistem kerja komputer memiliki *brainware*, *hardware*, dan *software*. Tanpa salah satu dari ketiga sistem tersebut, komputer tidak akan berguna. *Software* terbangun atas susunan program) dan *syntax* (cara penulisan atau pembuatan program). Untuk menyusun program atau *syntax*, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu *software*. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu *software*.⁴

Bahasa pemrograman komputer pertama yaitu bahasa mesin yang terdiri dari bilangan angka-angka, seiring berjalannya waktu bahasa pemrograman dikembangkan agar mempermudah untuk membuat suatu program. Bahasa pemrograman generasi kedua setelah bahasa mesin yaitu bahasa pemrograman *assembly*. bahasa pemrograman *assembly* adalah bahasa pemrograman tingkat rendah yang digunakan dalam pemrograman komputer, *mikroprosesor*, pengendali mikro, dan perangkat lainnya yang dapat diprogram. bahasa *assembly* mengimplementasikan representasi atas kode mesin dalam bentuk simbol-simbol yang secara relatif lebih dapat dipahami oleh manusia.⁵

³ Anna Nur Nazilah Chamim, "Penggunaan Microcontroller Sebagai Pendeteksi Posisi Dengan Menggunakan Sinyal GSM", *Jurnal Informatika* Vol. 4, No 1, (2017): 430, <https://media.neliti.com/media/publications/102952-ID-penggunaan-microcontroller-sebagai-pende.pdf>

⁴ Gun Gun Maulana, "pembelajaran dasar algoritma dan pemrograman Menggunakan EL-Algoritma berbasis WEB", *Jurnal Teknik Mesin*, Vol. 06 (2017):9, https://www.researchgate.net/publication/315547352_pembelajaran_dasar_algoritma_dan_pemrograman_menggunakan_elgoritma_berbasis_web

⁵ Indra Gunawan, *Pemrograman Assembly Mikroprosesor*, (Guepedia:2019), 16

Mikroprosesor lebih dikenal dengan sebutan *CPU* atau *Central Processing Unit* adalah sebuah rangkaian terintegrasi (*IC*) sebagai unit mesin pengolah yang bekerja melakukan fungsi pokok komputasi aritmatika dan logika. Fungsi *Mikroprosesor* adalah sebagai pengontrol atau pengolah utama dalam suatu rangkaian elektronik⁶

Bahasa pemrograman terus berkembang setiap tahunnya. perubahan teknologi telah menjadi katalis yang signifikan untuk pembuatan bahasa pemrograman. banyak para ilmuwan-ilmuan di berbagai negara seperti Jerman, Italia Inggris yang mengembangkan bahasa pemrograman. Disini peneliti tertarik untuk meneliti mengenai perkembangan bahasa pemrograman khususnya di Amerika Serikat, karena di Amerika Serikat banyak terdapat perusahaan – perusahaan perangkat lunak komputer, dan di Amerika Serikat juga diadakannya *Conference/Committee on Data Systems Languages (CODASYL)*.

Salah satu bahasa pemrograman dibuat oleh ilmuwan asal Amerika adalah bahasa pemrograman *Formula Translation (FORTRAN)*, bahasa pemrograman *FORTRAN* merupakan bahasa pemrograman tingkat tinggi pertama yang dibuat oleh John Backus, bahasa pemrograman *FORTRAN* merupakan bahasa pemrograman yang tertua dan masih digunakan hingga

⁶ Timbo Faritcan Parlaungan, Nita Wijaya, “perancangan perangkat lunak untuk simulasi pembuatan peta alamat memori pada sistem mikroprocessor” , *Jurnal Teknologi Informasi dan Komunikasi*, (2017):3, <https://jurnalstmiksubang.ac.id/index.php/jtik/article/view/47>

sekarang.⁷ Sebelum hadir *FORTRAN*, apabila seseorang akan memprogram komputer, maka harus menggunakan bahasa mesin yang sangat rumit.

Pada tahun 1950, seorang ahli dari perusahaan pabrik komputer yaitu *IBM (International Business Machine)* yang bernama John Backus⁸ John Backus berhasil mengembangkan suatu bahasa pemrograman komputer yang mudah untuk dipakai dan dipahami, bahkan oleh orang yang awam komputer sekalipun.

Bahasa pemrograman ini cukup mudah dan efektif untuk digunakan. Sehingga, bukan hal yang aneh apabila dengan cepat, bahasa ini berkembang di masyarakat. Bahasa pemrograman *FORTRAN* ditujukan sebagai aplikasi di bidang sains dan teknik. Namun pada saat ini, bahasa pemrograman *FORTRAN* harus bersaing dengan bahasa-bahasa pemrograman lain secara kompetitif.

Berdasarkan penjelasan di atas peneliti tertarik untuk melakukan penelitian dengan judul “**PERKEMBANGAN BAHASA PEMROGRAMAN DI AMERIKA SERIKAT TAHUN 1955 – 1995**”

B. Fokus Penelitian

Pada bagian fokus penelitian merupakan pengembangan dari uraian konteks penelitian yang menunjukkan bahwa penelitian yang telah ditelaah memang belum terjawab atau masih belum terpecahkan. Adapun fokus kajian yang akan disajikan oleh peneliti adalah sebagai berikut :

⁷ Muhammad Taufiq Pratama, “ Evolusi Bahasa pemrograman ”, *Jurnal Computech dan Bisnis*, Vol. 8 No. 1, (2014): 36, <https://www.jurnal.stmik-mi.ac.id/index.php/jcb/article/viewFile/112/136>

⁸ Ibid, 37

1. Bagaimana gambaran bahasa pemrograman yang muncul pada tahun 1955-1995 ?
2. Bagaimana dampak dari perkembangan bahasa pemrograman dalam perkembangan teknologi masa kini ?

C. Ruang Lingkup Penelitian

Dalam penelitian sejarah agar menghasilkan penelitian yang fokus penelitian ini mempunyai ruang lingkup yang berguna untuk membatasi objek penelitian. Hal ini bertujuan agar menghasilkan penelitian yang fokus dan tidak melebar dari tema besar yang diteliti. Pada penelitian ini terbagi dua ruang lingkup; lingkup spasial dan lingkup temporal.

Penelitian ini membatasi ruang lingkup spasial pada negara Amerika Serikat. Alasan wilayah tersebut yang saya pilih di negara Amerika Serikat karena di wilayah tersebut ada banyak sekali perusahaan perangkat lunak terbesar seperti IBM, Microsoft, Intel. Dan di Amerika Serikat juga diadakannya *Conference Committee on Data Systems Languages (CODASYL)*.

Pembatasan ruang lingkup temporal dalam penelitian ini berkisar dari tahun 1955-1995. Tahun 1955 menjadi batas awal penelitian ini karena pada tahun 1955 munculnya bahasa pemrograman tingkat tinggi pertama di Amerika Serikat dan terus menerus berkembang hingga tahun 1995.

D. Tujuan Penelitian

Tujuan penelitian merupakan gambaran tentang arah yang akan dituju dalam melakukan penelitian. Tujuan penelitian ini mengacu pada rumusan masalah yang telah dirumuskan. Adapun tujuan masalahnya adalah sebagai berikut :

1. Untuk mengetahui bagaimana gambaran bahasa pemrograman yang muncul pada tahun 1955-1995.
2. Untuk dampak dari perkembangan bahasa pemrograman dalam perkembangan teknologi masa kini.

E. Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan manfaat secara teoritis dan praktis

- a. Secara teoritis, penelitian ini diharapkan dapat memberikan sumbangan pemikiran dalam memperkaya wawasan keilmuan tentang bagaimana Perkembangan Bahasa Pemrograman Di Amerika Serikat Tahun 1955 – 1995.
- b. Secara praktis, penelitian ini diharapkan dapat menyumbangkan pemikiran terhadap pemecahan masalah yang berkaitan dengan Perkembangan Bahasa Pemrograman Di Amerika Serikat Tahun 1955 – 1995.

F. Studi Terdahulu

Penelitian terdahulu atau Studi terdahulu adalah kajian penelitian yang pernah dilakukan oleh peneliti sebelumnya yang dapat diambil dari berbagai sumber ilmiah seperti skripsi, tesis, disertasi atau jurnal penelitian. Berikut adalah penelitian terdahulu yang menjadi acuan peneliti dalam melakukan penelitian :

1. Penelitian yang dilakukan oleh Muhammad taufiq Pratama (2014)⁹, dalam penelitiannya dengan judul “Evolusi Bahasa pemrograman” hasil dari

⁹ Muhammad Taufiq Pratama “Sejarah Bahasa Pemrograman Dibuat,” Jurnal IMILKOM, (2021):13, <https://imilkom.usu.ac.id/?p=831>

penelitian ini adalah bahasa pemrograman, layaknya sebuah teknologi, selalu berubah dari waktu ke waktu untuk mengikuti perkembangan zaman. munculnya bahasa pemrograman baru setiap dekade, untuk memenuhi kebutuhan umat manusia yang semakin kompleks. Akibatnya, jumlah revisi maupun bahasa baru yang muncul tidak lagi dapat dihitung secara mudah, sehingga merekonstruksi sejarahnya menjadi hal yang tidak mudah. Untuk mengatasi hal ini, diadakanlah *Conference on the History of Programming Languages (HOPL)* sebanyak dua kali di Amerika Serikat pada tahun 1978 dan 1993.

Pada konferensi pertama, komite konferensi tersebut menentukan tiga belas bahasa pemrograman yang memenuhi kriteria berupa telah digunakan selama setidaknya sepuluh tahun sebelum diadakannya konferensi. Pada konferensi kedua bahasa pemrograman yang terpilih dipresentasikan oleh para pengembang dan bahasa pemrograman terpilih untuk selanjutnya didokumentasikan secara rinci, yang hasilnya kemudian menjadi dasar sejarah bahasa pemrograman yang diakui secara internasional. Namun tentu saja lingkup bahasa pemrograman yang terdokumentasi hanya bahasa pemrograman yang muncul sepuluh tahun sebelum konferensi kedua diadakan sehingga bahasa pemrograman yang muncul setelahnya tidak dicantumkan dalam dokumentasi HOPL.

Persamaan dalam penelitian ini adalah sama-sama membahas mengenai perkembangan bahasa pemrograman. Perbedaan dari penelitian ini adalah tahun pembahasannya. Pada penelitian ini membahas perkembangan bahasa

pemrograman dari tahun 1978 sampai dengan 1993 sedangkan penelitian yang akan dilakukan membahas mengenai Perkembangan Bahasa Pemrograman di Amerika Serikat Tahun 1955 – 1995.

2. Penelitian yang dilakukan oleh Kadek Wibowo, (2015)¹⁰, dengan Judul “Analisa Konsep *Object Oriented Programming* Pada Bahasa pemrograman *PHP*”, jenis penelitian yang digunakan oleh peneliti adalah metode penelitian studi kasus, hasil dari penelitian ini adalah Secara garis besar, bahasa pemrograman komputer adalah sebuah alat yang dipakai oleh para programmer komputer untuk menciptakan program aplikasi yang digunakan untuk berbagai macam keperluan.. Persamaan dari penelitian ini adalah sama sama membahas bahasa pemrograman. Perbedaan penelitian ini membahas mengenai bahasa pemrograman *PHP*, sedangkan penelitian yang akan dilakukan membahas mengenai Perkembangan bahasa pemrograman Di Amerika Serikat Tahun 1955 – 1995.

3. Penelitian yang dilakukan oleh Richy R. Saragih, (2018)¹¹, yang berjudul “pemrograman dan bahasa pemrograman” hasil dari penelitian ini adalah Program merupakan sebuah elemen inti untuk kinerja suatu perangkat, Program ini lah yang akan memproses perintah yang diberikan oleh user agar dapat dikelola oleh perangkat atau device, Program ini dibuat juga oleh manusia, dan orang yang membuat program ini disebut Programmer, Persamaan penelitian ini adalah sama – sama membahas bahasa

¹⁰ Kadek Wibowo “Analisa Konsep Object Oriented Programming Pada Bahasa pemrograman PHP”, *Jurnal khatulistiwa informatika*, Vol 3 No 2, (2015): 10, <https://ejournal.bsi.ac.id/ejurnal/index.php/khatulistiwa/article/view/1662>

¹¹ Richy R. Saragi, “ Program dan Bahasa Pemrograman”, *jurnal STMIK-STIE*, (2016):1-15.

pemrograman. Perbedaan dari penelitian ini adalah penelitian ini membahas bagaimana membuat suatu program, sedangkan penelitian yang akan dilakukan membahas Perkembangan bahasa pemrograman di Amerika Serikat Tahun 1955 – 1995.

4. Penelitian yang dilakukan oleh Lia Aulina, (2012), dengan judul “ Sejarah Bahasa pemrograman *Delphi* ” hasil dari penelitian ini adalah *Delphi* adalah sebuah bahasa pemrograman dan lingkungan pengembangan perangkat lunak *Delphi* awalnya sebuah proyek penelitian rahasia di Borland yang berevolusi menjadi sebuah produk yang disebut *AppBuilder*., Salah satu tujuan asli dari *Delphi* adalah untuk menyediakan konektivitas database untuk programmer sebagai kunci dan sebuah paket database yang populer pada saat itu adalah database *Oracle*. Persamaan dari penelitian ini adalah sama-sama membahas tentang perkembangan bahasa pemrograman. Perbedaannya pada penelitian ini adalah pada penelitian ini membahas sejarah bahasa pemrograman *Delphi*, sedangkan penelitian yang akan dilakukan membahas Perkembangan Bahasa Pemrograman Di Amerika Serikat Tahun 1955 – 1995.

5. Penelitian yang dilakukan oleh Effendi Dodi Arisandi¹², (2014), yang berjudul “ Kemudahan Pemrograman *Mikrokontroler Arduino* Pada Aplikasi Wahana Terbang ”, hasil dari penelitian ini adalah Bagi pelajar atau mahasiswa untuk mempelajari pemrograman *IC mikrokontroler* khususnya keluaran dari *ATMEL* dipermudah dengan kehadiran sistem *open source*

¹² Effendi Dodi Arisandi, “Kemudahan Pemrograman Mikrokontroler Arduino pada Aplikasi Wahana terbang”, *Jurnal Untirta*, Vol 3 No 2 (2014):10-15 <https://jurnal.untirta.ac.id/index.php/jis/issue/view/108>.

arduino, jika dibandingkan dengan keadaan 10 tahun yang lalu, pemrograman *mikrokontroler* pada umumnya masih menggunakan bahasa mesin yang lebih populer disebut dengan bahasa *assembler* dan cukup sulit untuk memahaminya, dari hasil kajian penggunaan *system arduino* dapat disimpulkan bahwa dengan *system open source* dan berbiaya murah maka perkembangan teknologi elektronika khususnya di bidang aplikasi *mikrokontroler* dapat berkembang secara pesat, teknologi pembuatan sebuah *IC* dapat dikelompokkan berdasarkan jumlah transistor pada *IC*.

Pemrograman mikrokontroler diawali dengan penggunaan bahasa mesin yang lebih populer disebut dengan bahasa *assembler*, bahasa *assembly* atau bahasa mesin memerlukan pemahaman yang sangat mendalam dikarenakan berhubungan langsung dengan hardware-nya, perubahan operating sistem pada komputer sangat mempengaruhi perkembangan bahasa pemrograman pada *mikrokontroler*, pada saat ini ada beberapa macam bahasa pemrograman untuk *mikrokontroler* seperti *basic* dengan *editor* dan *compiler bascom*, *c++*, *mikro pascal*.

Persaingan pasar bagi industri *mikro controller* sangat dipengaruhi oleh kemudahan pemrograman *mikrokontroler* tersebut beserta fungsi-fungsi pendukungnya. Pada saat ini sangat berkembang bahasa pemrograman yang berbasis *open source*. Dengan keterbukaan dari inti bahasa pemrograman suatu *mikrokontroler* maka bahasa pemrograman tersebut akan dapat berkembang dengan pesat. Persamaan dari penelitian ini adalah sama-sama membahas tentang perkembangan bahasa pemrograman. Perbedaannya pada

penelitian ini adalah pada penelitian ini membahas manfaat serta bagaimana cara membuat suatu program, sedangkan penelitian yang akan dilakukan membahas perkembangan bahasa pemrograman yang muncul tahun 1940-1990.

G. Kerangka Konseptual

Perkembangan bahasa pemrograman

Perkembangan bahasa pemrograman adalah suatu proses kegiatan dalam rangka mengembangkan bahasa pemrograman atau ilmu tentang keterampilan pemrograman. Bahasa pemrograman, atau sering diistilahkan juga dengan bahasa komputer atau bahasa pemrograman komputer, adalah instruksi standar untuk memerintah komputer.¹³ Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

Bahasa Pemrograman (*programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar menjalankan fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan,

¹³ Linda W. Friedman, "sejarah singkat bahasa pemrograman", compulang 1991.

dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi. Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

1. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa *biner*, contohnya 01100101100110.
2. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bahasa *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode mnemonic), contohnya kode mesin *MOV*, *SUB*, *CMP*, *JMP*, *JGE*, *JL*, *LOOP*.
3. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <, &&, ||.
4. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya begin, end, if, for, while.

Komputer dapat mengerti bahasa manusia itu diperlukan program compiler atau interpreter. Sebagian besar bahasa pemrograman digolongkan sebagai Bahasa Tingkat Tinggi, hanya bahasa C yang digolongkan sebagai Bahasa Tingkat Menengah dan Assembly yang merupakan Bahasa Tingkat Rendah.

Komputer merupakan sebuah mesin yang dirancang untuk mengikuti instruksi. Program komputer merupakan sebuah instruksi yang digunakan oleh komputer untuk memecahkan masalah atau tugas-tugas yang diberikan padanya.

Misalnya jika menggunakan komputer untuk menghitung pembayaran pada seorang pegawai maka Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Menampilkan pesan pada layar “ berapa jam kamu bekerja ?”
2. Menunggu pengguna memasukan data jam bekerja dan setelah ditekan ENTER maka komputer akan memasukan data ke dalam memory.
3. Menampilkan pesan pada layar “berapa besarnya gaji tiap jamnya?”
4. Menunggu pengguna memasukan data per jam bekerja dan setelah ditekan ENTER maka komputer akan memasukan data ke dalam memory.
5. Komputer melakukan perkalian antara jumlah jam dengan gaji perjam.
6. Komputer menampilkan hasil perhitungan gaji pegawai yang harus dibawa pulang.

Sekumpulan instruksi-instruksi tersebut disebut dengan algoritma. Algoritma merupakan kumpulan instruksi yang terdefinisi langkah demi langkah secara baik dengan tujuan untuk menyelesaikan masalah. Catatan langkah-langkah ini diurutkan secara sekuensial. Dalam algoritma langkah pertama harus dilakukan lebih dahulu sebelum langkah kedua dan seterusnya.

H. Metode Penelitian

Adapun metode yang digunakan oleh peneliti adalah metode penelitian sejarah. Dalam pembelajaran metode penelitian sejarah, sering dikenal dengan istilah Historiografi. Kata “ Historiografi dapat dimaknai sebagai hasil atau karya dalam penulisan sejarah. Juga bisa dikatakan sarana mengkomunikasikan hasil-hasil penelitian yang diungkap, diuji (verifikasi), dan Di interpretasi. Berdasarkan pengertian ini dapat dipahami bahwa peristiwa sejarah memerlukan

penelitian sebelum disajikan dalam bentuk historiografi. Historiografi dapat dihasilkan melalui penelitian sejarah. Sama halnya dengan penelitian ilmiah lainnya. Penelitian sejarah juga memiliki tahapan metode penelitian.

1. Pemilihan topik

Pemilihan topik menjadi urutan pertama dalam penelitian sejarah Menurut Kuntowijoyo dikarenakan topik yang akan dijadikan penelitian sejarah itu cukup banyak sehingga penting bagi sejarawan untuk menemukan topik terlebih. ¹⁴

2. Heuristik (Pengumpulan Sumber).

Heuristik adalah proses mencari dan menemukan sumber yang diperlukan. Heuristik adalah teknik yang bisa membantu peneliti untuk memperoleh sumber yang dibutuhkan dalam penelitian. Data penelitian ini diperoleh melalui telaah pustaka dengan mengumpulkan buku-buku yang berkaitan dengan topik penelitian ini. sumber data yang akan peneliti gunakan yaitu :

a. Sumber Primer

Yang dimaksud dengan sumber primer adalah bila sumber atau penulis sumber menyaksikan, mendengar sendiri atau mengalami sendiri peristiwa yang dituliskan dalam sumber tersebut.¹⁵ Dalam skripsi ini sumber primer yang digunakan oleh peneliti adalah buku karya Jean E. Sammet yang berjudul *Programming Languages: History and*

¹⁴ Kuntowijoyo, *Pengantar Ilmu Sejarah*, (Yogyakarta: PT Bentang Pustaka 2005), 100

¹⁵ . Nina Herlina, *Metode Sejarah* , (Bandung: Satya Historika 2020), 24.

Fundamentals (New York: Prentice-Hall, 1969). Alasan peneliti menggunakan buku karya Jean E. Sammet. Karena Jean E. Sammet terlibat dalam sejarah pembuatan bahasa pemrograman di Amerika Serikat. Jean E. Sammet menggunakan bukunya untuk mengadvokasi bahasa tingkat tinggi pada saat bahasa rakitan populer dan ada keraguan luas tentang nilai bahasa tingkat tinggi di bidang pemrograman.

b. Sumber Sekunder

Selanjutnya peneliti juga menggunakan beberapa sumber sekunder. Sugiyono menerangkan sumber sekunder adalah sumber yang tidak langsung memberikan data kepada pengumpul, bisa lewat orang lain atau dokumen yang ditulis oleh orang lain.¹⁶ dalam penelitian ini sumber sekunder merupakan buku-buku penunjang yang berhubungan dengan persoalan yang dibahas. Data sekunder ini berfungsi sebagai pelengkap data primer yang digunakan dalam penelitian ini. Sumber data sekunder yang digunakan peneliti, yaitu:

1. History of Programming Languages karya Richard G. Gibson, Thomas J. Bergin (New York : ACM Press Staff 1996).
2. Bahasa Pemrograman karya Zinsari (Yogyakarta : Andi Offset., 1986).
3. History of Programming Languages karya Richard L. Wexelblat (New York: Academic Press 1981).

¹⁶ Sugiyono, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*, (Bandung: Afabeta 2011), 308.

4. Concepts of Programming Languages karya Robert W. Sebesta (New York : Addison Wesley 2005).

3. Kritik sumber/verifikasi sumber.

Dalam hal ini dilakukan uji keabsahan sumber (auntentisitas) yang dilakukan melalui kritik ekstern dan keabsahan tentang kesahihan sumber (kredibilitas) yang ditelusuri melalui kritik intern. Interpretasi (penafsiran) Setelah peneliti mengumpulkan sumber, kemudian memverifikasi atau melakukan kritik sumber,

4. Interpretasi (analisis)

Langkah selanjutnya yakni melakukan interpretasi (analisis) terhadap sumber yang telah dianalisis tersebut. Interpretasi sejarah sering disebut juga dengan analisis sejarah atau penafsiran data dan fakta sejarah..

5. Historiografi

yang terakhir yaitu Historiografi atau penulisan sejarah, yaitu proses penyusunan hasil penelitian yang telah diperoleh sehingga menjadi satu kesatuan yang utuh dalam bentuk skripsi, sehingga dihasilkan suatu tulisan yang logis dan sistematis, dengan demikian akan diperoleh suatu karya ilmiah yang dapat dipertanggungjawabkan kebenarannya.

I. Sistematika Pembahasan

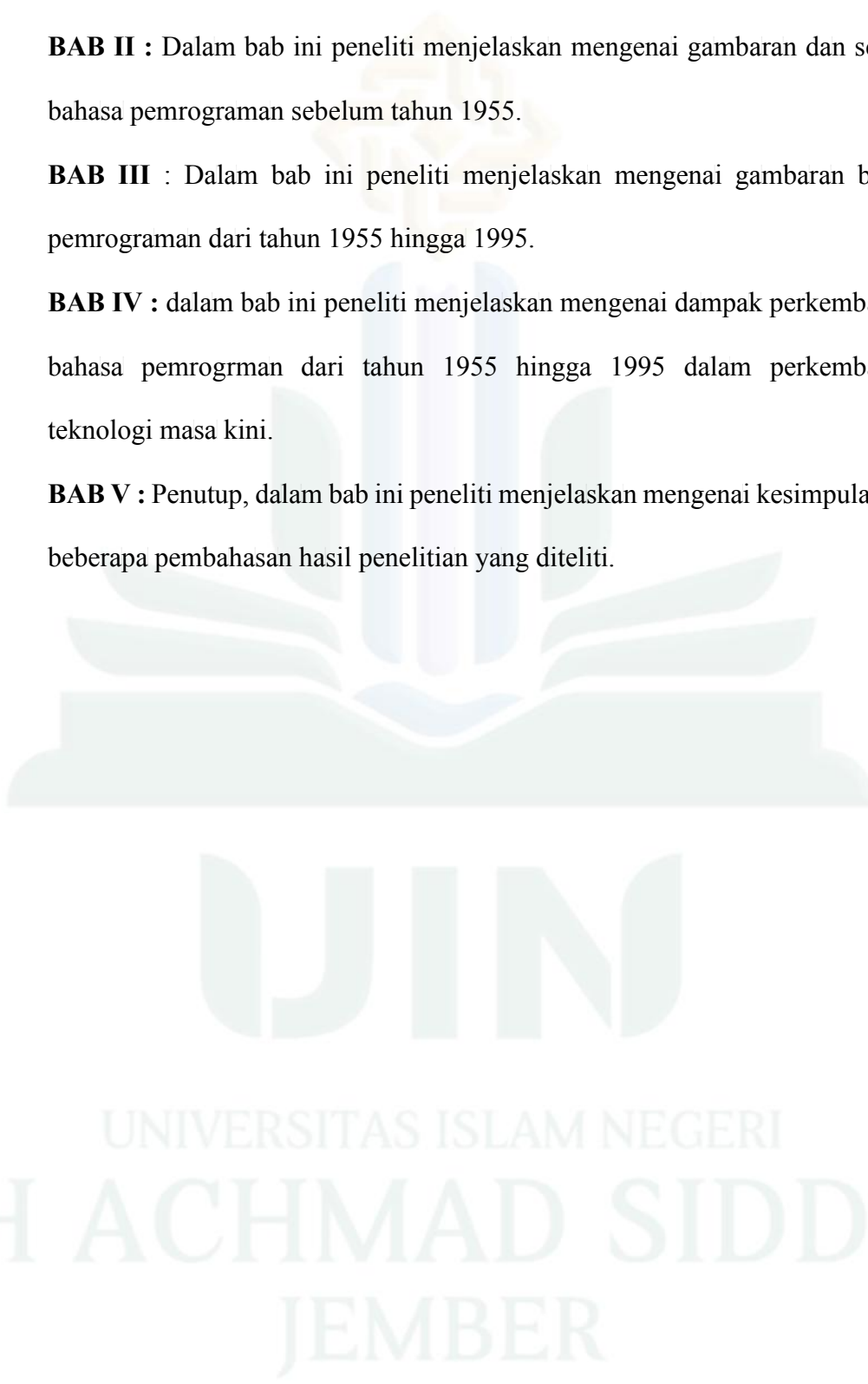
BAB I : Dalam bab ini peneliti menjelaskan mengenai konteks penelitian, fokus Penelitian, ruang lingkup penelitian, tujuan penelitian, manfaat penelitian, studi terdahulu, kerangka konseptual, metode penelitian, dan sistematika pembahasan.

BAB II : Dalam bab ini peneliti menjelaskan mengenai gambaran dan sejarah bahasa pemrograman sebelum tahun 1955.

BAB III : Dalam bab ini peneliti menjelaskan mengenai gambaran bahasa pemrograman dari tahun 1955 hingga 1995.

BAB IV : dalam bab ini peneliti menjelaskan mengenai dampak perkembangan bahasa pemrograman dari tahun 1955 hingga 1995 dalam perkembangan teknologi masa kini.

BAB V : Penutup, dalam bab ini peneliti menjelaskan mengenai kesimpulan dari beberapa pembahasan hasil penelitian yang diteliti.



BAB II

SEJARAH BAHASA PEMROGRAMAN SEBELUM 1955

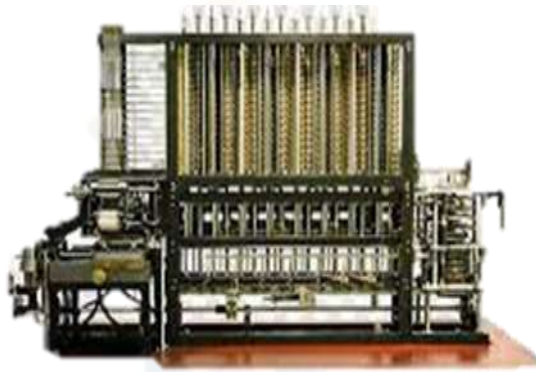
A. Sejarah munculnya bahasa pemrograman

Terciptanya bahasa pemrograman beriringan dengan sejarah mesin dan komputer. Pada masa itu, perhitungan dengan menggunakan tabel matematika sering mengalami kesalahan. Babbage ingin mengembangkan cara melakukan perhitungan secara mekanik, sehingga dapat mengurangi kesalahan perhitungan yang sering dilakukan oleh manusia. Pada saat itu, Babbage mendapat inspirasi dari perkembangan mesin hitung yang dikerjakan oleh Wilhelm Schickard, Blaise Pascal, dan Gottfried Leibniz.

Sejarah Komputer diawali ketika Penemu Inggris Charles Babbage menyelesaikan prinsip-prinsip pemakaian umum komputer digital seabad penuh sebelum perkembangan besar-besaran mesin hitung elektronik terjadi. Mesin yang dirancangnya, Awal mula bahasa pemrograman dimulai dari *Antikythera* yang berasal dari Yunani kuno.¹ *Antikythera* adalah kalkulator yang menggunakan beberapa tuas dan konfigurasi untuk menjalankannya.

Bahasa pemrograman pertama adalah algoritma yang dibuat oleh Ada Lovelace pada tahun 1843. Ada Lovelace menciptakan algoritma ini untuk mesin *Difference Engine* Charles Babbage. Tujuan dari algoritma ini adalah untuk menghitung bilangan Bernoulli, yang diakui oleh beberapa sejarawan sebagai program komputer pertama yang diterbitkan di dunia.

¹ Muhammad Taufiq Pratama “Sejarah Bahasa Pemrograman Dibuat,” *Jurnal IMILKOM*, (2021):13, <https://imilkom.usu.ac.id/?p=831>



Gambar 2. 1 mesin *Difference Engine*

Sumber: Ruang Guru, Minggu 31 Oktober 2021

Ada Lovelace menerjemahkan hasil karya matematikawan Italia *Luigi Menabrea* yaitu Mesin *Difference Engine* Charles Babbage ke dalam bahasa Inggris². Mesin tersebut akan menggunakan kartu berlubang tempat informasi akan dikodekan. Dia merencanakan metode untuk menghitung angka Bernoulli pada kartu-kartu tersebut. Ada Lovelace merincikan mesin penghitung yang mencakup elemen-elemen berikut:

1. Perangkat untuk mengkodekan informasi digital.
2. Memori untuk merekam nilai-nilai numerik.
3. Unit kontrol bagi pengguna untuk menunjukkan tugas mana yang akan dijalankan mesin.
4. Disk untuk menyelesaikan kalkulasi.
5. Perangkat yang memungkinkan Anda melihat hasilnya.

Pada awalnya Mesin *Difference Engine* Charles Babbage hanya bisa dibuat untuk menjalankan tugas-tugas dengan mengubah gigi yang dieksekusi perhitungan. Dengan demikian, bentuk paling awal dari sebuah bahasa komputer

² Abrahams Paul, "History Of Programming", American Mathematical Society, <http://www.jstor.org/stable/2004521>

adalah gerakan fisik. Akhirnya, gerakan fisik digantikan oleh sinyal-sinyal listrik ketika Pemerintah Amerika Serikat membangun *ENIAC* pada tahun 1942.

Hal tersebut diikuti banyak dari prinsip-prinsip yang sama dari Mesin *Difference Engine* Charles Babbage dan oleh karena itu, hanya dapat diprogram contohnya *switch* dan mempengaruhi seluruh sistem untuk masing-masing program baru atau perhitungan. Proses ini terbukti menjadi sangat membosankan.

Pada tahun 1945, John Von Neumann bekerja di *Institute for Advanced Study*.³ Dia mengembangkan dua konsep penting yang secara langsung mempengaruhi jalur pemrograman bahasa komputer. Yang pertama dikenal sebagai *shared program teknik*. teknik yang menyatakan bahwa sebenarnya perangkat keras komputer harus sederhana dan tidak perlu *hand wired* untuk masing-masing program. Sebaliknya, instruksi yang kompleks harus digunakan untuk mengontrol perangkat keras sederhana, yang memungkinkan untuk memprogram jauh lebih cepat.

Konsep kedua yang juga sangat penting untuk perkembangan bahasa pemrograman. Von Neumann menyebutnya *conditional control transfer*. Ide ini memunculkan gagasan dari subrutin, atau blok kecil dari kode yang bisa melompat ke dalam urutan apapun, bukan satu set diurutkan secara kronologis langkah-langkah untuk komputer untuk mengambil. Bagian kedua dari ide yang menyatakan bahwa kode komputer harus mampu cabang berdasarkan logis pernyataan-pernyataan seperti: IF (ekspresi) kemudian, dan dilingkarkan seperti dengan

³ Richard I. Wexelblat, *History of Programming Languages*, (New York: Academic Press, 1981), 80.

pernyataan *FOR*. "*Conditional control transfer*" memunculkan ide "perpustakaan", yang merupakan blok dari kode yang dapat digunakan kembali berulang.

Pada tahun 1949, beberapa tahun setelah Von Neumann bekerja, bahasa Pendek Kode yang muncul Ini adalah pertama bahasa komputer untuk perangkat elektronik dan dibutuhkan programmer untuk mengubah pernyataan menjadi 0 dan 1 dengan tangan. Namun, hal tersebut adalah langkah pertama menuju bahasa yang kompleks saat ini. Pada tahun 1951, Grace Hopper menulis *compiler*, *Compiler* adalah program yang mengubah bahasa laporan ke 0 dan 1 untuk komputer untuk mengerti. menyebabkan lebih cepat pemrograman, sebagai programmer tidak lagi harus melakukan pekerjaan dengan tangan.

Pada tahun 1940, komputer bertenaga listrik modern pertama yang dapat dikenali diciptakan. Kecepatan dan kapasitas memori yang terbatas memaksa program untuk menulis program bahasa rakitan yang disetel dengan tangan.⁴ Akhirnya disadari bahwa pemrograman dalam bahasa *assembly* membutuhkan banyak usaha intelektual. Proposal awal untuk bahasa pemrograman tingkat tinggi adalah *Plankalkül*, yang dikembangkan oleh Konrad Zuse untuk komputer Z1 antara tahun 1943 dan 1945 tetapi tidak diimplementasikan pada saat itu.

B. Bahasa pemrograman tingkat rendah

Bahasa pemrograman tingkat rendah adalah bahasa pemrograman yang menggunakan set instruksi khusus dari sebuah prosesor yang sudah ditentukan oleh

⁴ Donald E. Knut, "The Early Development Of Programming Languages", (*Skripsi*, Stanford University, California 1976), 10.

pembuatnya. Artinya, bahasa pemrograman tingkat rendah hanya dapat dijalankan pada prosesor tertentu yang ditujukan ketika bahasa ini dibuat. Untuk alasan tersebut, kode yang ditulis dalam bahasa pemrograman tingkat rendah tidak portabel dan tidak akan berjalan pada komputer dengan prosesor yang berbeda. Pada tahun 1948, Konrad Zuse mempublikasikan sebuah paper tentang bahasa pemrograman miliknya yakni Plankalkül. Bagaimanapun, bahasa tersebut tidak digunakan pada masanya dan terisolasi terhadap perkembangan bahasa pemrograman yang lain.⁵



Gambar 2.2 Mesin Komputer Z1 di buat ole Konrad Zuse
Sumber: WordPress, Kamis 29 Januari 2015

karakteristik dari bahasa pemrograman tingkat rendah:

1. Memiliki hubungan 1:1 dengan instruksi prosesor.
2. Tidak portable (hanya bekerja pada prosesor tertentu).
3. Memungkinkan kontrol individu atas komponen dan register prosesor.
4. Program biasanya membutuhkan lebih sedikit memori dan dapat dieksekusi lebih cepat daripada program pada bahasa pemrograman tingkat tinggi yang melakukan tugas yang sama.

⁵ Richard I. Wexelblat, *History of Programming Languages*, (New York: Academic Press, 1981), 80.

Berikut contoh fungsi dan sistem bahasa pemrograman tingkat rendah :

a. Sistem Tertanam (*Embedded System*)

Embedded system adalah sistem komputer lengkap yang memiliki fungsi khusus, seperti jam tangan digital, peralatan rumah tangga, dan sistem pemanas sentral. Sistem ini dirancang untuk menyelesaikan sejumlah tugas tertentu dengan sangat efisien. *Embedded system* biasanya akan memiliki ukuran yang relatif kecil. Oleh karena itu, sistem ini tidak memiliki prosesor besar yang memerlukan sistem pendinginan. Artinya, program yang berjalan pada sistem ini harus dapat bekerja dengan efisien dan memanfaatkan sumber daya yang ada dengan sebaik-baiknya.

b. *Device Driver*

Device driver adalah program yang memungkinkan komputer untuk berinteraksi dengan periferal, seperti mouse atau printer. Sistem operasi akan menginstal dan mengelola *device driver* ini untuk setiap periferal yang terhubung ke komputer.

Fungsi utama *device driver* adalah sebagai penghubung utama hardware. *device driver* tidak akan berguna tanpa hardware begitu juga Sebaliknya. Fungsi dari *device driver* lainnya adalah:

1. penghubung *Software* dengan *hardware*.
2. Penerjemah data antara *Software* dan *hardware*.
3. Pengontrol *Input* dan *Output Software* terhadap *hardware*.

Pada tahun 1949, Kalkulator Otomatis Penyimpanan Penundaan Elektronik adalah yang pertama menggunakan bahasa rakitan, yang dianggap sebagai

bahasa pemrograman komputer tingkat rendah yang memecah bahasa kompleks kode mesin. Ini berarti bahwa instruksi untuk mengoperasikan komputer dimasukkan ke dalam istilah yang lebih sederhana.

Pada tahun 1952, Alick Glennie menciptakan istilah *Autocode* yang berarti “keluarga bahasa pemrograman.” Dia adalah seorang ilmuwan komputer yang awalnya menggunakan *autocode* untuk komputer Mark 1 di University of Manchester di Inggris.⁶ *Autocode* dianggap sebagai bahasa pemrograman komputer pertama yang dikompilasi, artinya dapat diterjemahkan langsung ke dalam kode mesin menggunakan program *compiler*. Jadi, bahasa mesin dan bahasa rakitan termasuk dalam kelas bahasa pemrograman tingkat rendah.



Gambar 2.3 Komputer Mark 1

Sumber: The University of Manchester 1999

Bahasa pemrograman pertama berfungsi yang dirancang untuk mengkomunikasikan instruksi ke komputer ditulis pada awal 1950. Kode Pendek John Mauchly, diusulkan pada tahun 1949, adalah salah satu bahasa tingkat tinggi pertama yang pernah dikembangkan untuk komputer elektronik⁷. Tidak seperti

⁶ Heny Ledgord, *The World of Programming Languages* (New York: Science Research Associates, 1986), 45.

⁷ Alice E. Fischer, *The Anatomy of Programming Languages*, (New York: : Prentice Hall, 1993), 51

kode mesin, pernyataan Kode Pendek mewakili ekspresi matematika dalam bentuk yang dapat dimengerti. Namun, program harus diterjemahkan ke dalam kode mesin setiap kali dijalankan, membuat prosesnya jauh lebih lambat daripada menjalankan kode mesin yang setara.

Kode mesin (bahasa mesin) adalah sistem instruksi untuk komputer tertentu, yang ditafsirkan langsung oleh prosesor komputer tersebut, didasarkan pada seperangkat perintah dasar yang dapat dimengerti oleh prosesor. menulis data ke sel memori yang diinginkan, menambahkan data dari *register prosesor* yang berbeda. setiap instruksi adalah bilangan biner, tetapi untuk mempermudah, perintah juga dapat ditulis dalam notasi heksadesimal, dan baru kemudian diterjemahkan ke dalam kode biner.

Program dalam bentuk kode mesin terlihat sangat rumit, misalnya, program untuk salah satu prosesor modern yang menampilkan pesan "Halo, dunia!" terlihat seperti ini dalam bahasa mesin: BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD 20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21. Tidak mungkin untuk tidak setuju dengan fakta bahwa tidak hanya ini rumit, tetapi juga membutuhkan pengetahuan khusus untuk persepsi.

Perlu ditekankan bahwa setiap jenis prosesor dapat memiliki serangkaian instruksinya sendiri yang tidak tumpang tindih dengan jenis lainnya. Oleh karena itu, program kode mesin untuk prosesor A terlihat satu arah, tetapi program yang melakukan hal yang persis sama mungkin terlihat sangat berbeda untuk prosesor B.

Dari sini dapat disimpulkan bahwa kode program praktis tidak portabel dari satu mesin ke mesin lainnya, yang mengarah pada penulisan dari awal.⁸

Jadi, untuk memprogram komputer, perlu mengetahui bahasa mesin dan memahami perangkat komputer yang digunakan untuk menulis program. Singkatnya, ini menetapkan persyaratan tinggi untuk programmer pada waktu itu, oleh karena itu profesi itu sendiri tidak ada, dan programmer adalah ahli matematika dan fisikawan yang menggunakan komputer untuk perhitungan yang diperlukan.

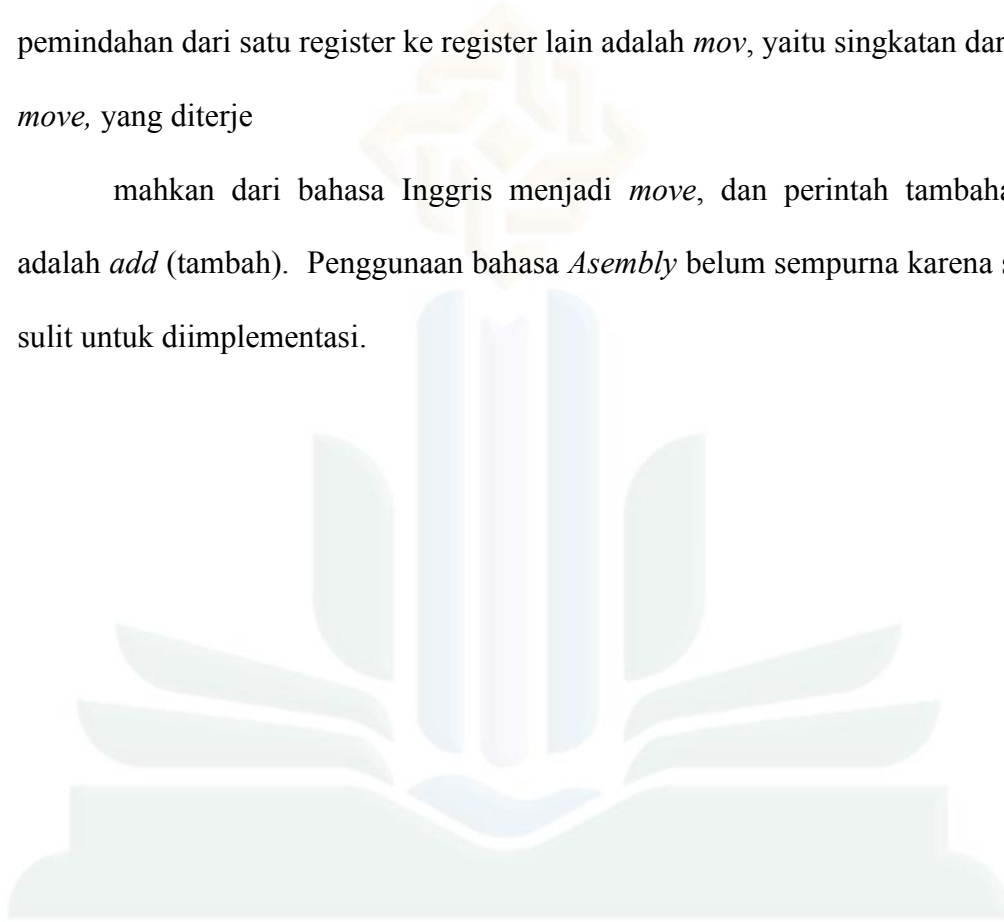
Pada awal paruh kedua abad ke XX permintaan untuk pengembangan perangkat lunak meningkat, bersamaan dengan ini, program menjadi lebih rumit, masing-masing, dan kode sumbernya menjadi semakin kompleks. Ini mendorong programmer untuk menggunakan program khusus, *assembly*, yang memungkinkan untuk menerjemahkan kode program dalam bahasa rakitan ke dalam kode mesin. Transisi ke penggunaan *assembly* dapat dianggap sebagai tahap selanjutnya dalam pengembangan bahasa pemrograman.

Bahasa *assembly* adalah format tertentu untuk menulis instruksi mesin yang nyaman untuk persepsi manusia. Artinya, sekarang pengembang mungkin tidak tahu seperti apa perintah prosesor dalam bentuk numerik, ia hanya perlu mengetahui catatan instruksi dalam bahasa *assembly*. Selain itu, catatan ini lebih mudah diingat, yang berarti bahwa keuntungan berikut telah muncul: program menjadi lebih mudah untuk ditulis; mereka terlihat lebih mudah dibaca dan dimengerti oleh seseorang daripada dalam kode mesin. Misalnya, perintah

⁸ Mutiara Ramadya Ramadya Kusuma, "Evolution of (Evolution of Programming Language)", Scribid, 2010: 6. <https://id.scribd.com/doc/65375992/Evolusi-Bahasa-Pemrograman>.

pemindahan dari satu register ke register lain adalah *mov*, yaitu singkatan dari kata *move*, yang diterje

mahkan dari bahasa Inggris menjadi *move*, dan perintah tambahannya adalah *add* (tambah). Penggunaan bahasa *Assembly* belum sempurna karena selain sulit untuk diimplementasi.



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

BAB III

PERKEMBANGAN BAHASA PEMROGRAMAN TAHUN 1955 - 1995

A. Bahasa Pemrograman Tingkat Tinggi

Pada tahun 1951, sebuah *compiler* generasi pertama diciptakan. Kompiler tersebut bernama A-0. Penciptanya adalah Grace Murray Hopper. Semuanya bermula pada tahun 1949 ketika Grace bekerja pada *Eckert-Mauchly Computer Corporation* sebagai matematikawan senior. Selama bekerja, Grace merancang sebuah *compiler*, yaitu perangkat yang menerjemahkan bahasa pemrograman yang dimengerti manusia, menjadi bahasa yang dimengerti komputer.¹ Motivasi Grace dalam menciptakan *compiler* sungguh di luar dugaan. Grace mengaku bahwa seorang yang malas, karena itu diciptakannya lah sebuah *compiler* agar pekerjaan pemrograman menjadi lebih mudah.

Pada tahun 1957, Grace bersama dengan divisi yang dipimpinnya menelurkan kompiler A-0 generasi baru yang diberi nama *Math-Matic*. *Math-Matic* adalah compiler pertama yang menggunakan bahasa Inggris. Karya-karya Grace berikutnya adalah A-2, A-3, dan AT-3. Pada masa kini, kompiler A-2 disebut dengan Assembly, sedangkan AT-3 mirip dengan Fortran.

Untuk memenuhi bisnis saat itu, *The Remington Rand Corporation* memproduksi sebuah bahasa pemrograman bernama *COBOL* (*Common Business Oriented Language*). Sudah tentu bahasa tersebut adalah hasil sumbangsiah Grace juga. bahasa pemrograman *COBOL* digunakan dalam pemrosesan kartu, ATM, dan

¹ Donald E. Knut, "The Early Development Of Programming Languages", (*Skripsi*, Stanford University, California 1976), 10.

bahkan digunakan dalam film *Terminator* untuk tampilan visual *Terminator*. *COBOL* merupakan bahasa pemrograman yang didedikasi untuk keperluan bisnis, keuangan, dan sistem administrasi.² Sedangkan keperluan sains, bahasa pemrograman yang diciptakan adalah *Fortran* (*Formula Translator*) yang dikembangkan oleh *IBM*.

Bahasa pemrograman *Fortran* telah mengandung *IF*, *DO*, dan *GOTO* yang merupakan loncatan besar dalam perkembangan bahasa pemrograman. Pada tahun 1958, *IBM* merilis versi bahasa yang direvisi, bernama *FORTRAN II*. memberikan dukungan untuk pemrograman prosedural dengan memperkenalkan pernyataan yang memungkinkan pemrogram membuat subrutin dan fungsi, sehingga mendorong penggunaan kembali kode.

Popularitas *FORTRAN* yang semakin meningkat membuat banyak produsen komputer mengimplementasikan versinya untuk mesin mereka sendiri. Setiap pabrikan menambahkan penyesuaiannya sendiri, sehingga tidak mungkin untuk menjamin bahwa program yang ditulis untuk satu jenis mesin akan dikompilasi dan dijalankan pada jenis yang berbeda. *IBM* merespons dengan menghapus semua fitur yang bergantung pada mesin dari versi bahasanya. Hasilnya, dirilis pada tahun 1961, disebut *FORTRAN IV*.

Pada tahun 1958, seorang ilmuwan bernama *John McCarthy* dari *MIT* menciptakan sebuah bahasa pemrograman bernama *Lisp* (*List Processing*). Bahasa pemrograman *Lisp* di desain untuk keperluan riset, desain pesawat Grammarly,

² James J. Horning, *History of Programming Languages II*, (New York: ACM SIGPLAN, 1976) 34.

online text, dan layanan pengolahan. pembeda *Lips* dengan bahasa pemrograman lain³, Adalah pada tipe datanya. Pada *Lips*, satu-satunya tipe data yang dikenal adalah *list* (daftar), ditandai oleh serangkaian item yang diapit oleh tanda kurung (*parenthesis*).

Lips sendiri ditulis dengan serangkaian *List*, jadi *Lips* memiliki kemampuan unik untuk dapat berkembang dari dirinya sendiri. Sintaks *Lips* dikenal sebagai *Cambridge polish* dan sangat berbeda dari logika boolean standar. Karena spesifik dan bersifat abstrak, maka *Lips* masih digunakan hingga kini. Versi modern *Lips* bahkan sudah mendukung pemrograman berorientasi objek, kemampuan untuk mengakses *database*, memiliki *IDE (Integrated Development Environment)* berbentuk grafis.

Pada tahun 1960, seorang matematikawan bernama Kenneth Iverson mengembangkan bahasa pemrograman *APL (A Programming Language)*. *APL* merupakan interpreter yang spesifik untuk masalah matematis serta memiliki sintaks dan karakter-karakter yang aneh. Pada tahun yang sama, rilis pula bahasa pemrograman *ALGOL*. Bahasa pemrograman *ALGOL* dikembangkan oleh sebuah komite sains.

ALGOL merupakan bahasa pemrograman pertama yang menggunakan struktur blok. *ALGOL* bahkan disebut-sebut sebagai nenek moyang dari bahasa *Pascal*, *C*, *C++* dan *Java*. Ada beberapa rilis dari bahasa pemrograman *ALGOL*

³ James J. Horning, *History of Programming Languages*, (New York: ACM SIGPLAN, 1976) 17.

dan masing-masing diberi nama sesuai tahun rilisnya, yaitu: ALGOL 58, ALGOL 60, dan ALGOL 68.

Bahasa *Snobol* diciptakan oleh tiga orang yang bekerja di laboratorium Bell. Mereka adalah D.J.Farber, R.E.Griswold, dan F.P.Polensky. *snobol* merupakan akronim dari *String Oriented Symbolic Language*. Dari namanya ketahuan bahwa Snobol lebih digunakan untuk keperluan pemrograman yang berhubungan dengan *string* (teks). Salah satu aplikasi pertama yang dibuat dengan *snobol* adalah *editor text*. Produk yang mengekor Snobol adalah *Fasbol (Compiler SNOBOL)*, dan *Spitbol (Speedy Implementation of Snobol)*.

Pada tahun 1963, IBM mengembangkan *PL 1 (Programming Language 1)*. IBM bermaksud mengombinasikan kelebihan-kelebihan *Algol*, *Fortran*, dan *Cobol* ke dalam suatu bahasa.⁴ Akibatnya bahasa *PL1* sangat berkompeten menjadi bahasa pemrograman yang sangat kompleks. Bagusnya, *PL 1* sangat berkompeten dalam membangun berbagai aplikasi. Namun, disisi lain *compiler* yang dibutuhkan sangat kompleks dan melakukan debugging pada *PL 1*.

BASIC (Beginner All Purpose Symbolic Instruction Code) diciptakan oleh John Kennedy dan Thomas Kurtz pada tahun 1964. Tidak seperti pencipta bahasa pemrograman yang lain yang umumnya ahli matematika atau ilmunan di bidang lain, John Kennedy dan Thomas Kurtz adalah baru saja lulus kuliah. Karena itu, tidak mengherankan jika bahas ciptaan mereka yang sangat simple. Bahasa pemrograman *BASIC* memang ditujukan bagi mereka yang bukan ahli di bidang computer. Tidak heran bila *BASIC* merupakan bahasa yang sangat populer, bahkan hingga kini.

⁴ James J. Horning, *History of Programming Languages*, (New York: ACM SIGPLAN, 1976) 10.

Bahasa pemrograman *BASIC* yang sekarang bukanlah basic yang dulu, tetapi variannya seperti *Visual Basic*, *VBScripts*, dan *Visual Basic for Application (VBA)*. Pada saat pertama kali diciptakan, *BASIC* bersifat *compiler*, namun kebanyakan *BASIC* yang ada sekarang adalah *interpreter*.

Pada tahun 1966 ada dua bahasa pemrograman yang diciptakan, yaitu *FORTRAN 66* yang merupakan pengembangan *FORTRAN IV (1961)* serta bahasa pemrograman *LOGO*.⁵ *LOGO* diciptakan untuk mengajarkan prinsip-prinsip pemrograman dan pemecahan masalah (*problem solving*) bagi anak-anak. Karena ditujukan untuk anak-anak, *LOGO* menggunakan bentuk visual, yaitu sosok tokoh seekor kura-kura. Pada prinsipnya, pemrograman *Logo* adalah memerintahkan suatu pola atau gambar.

pada tahun 1967, Marin Richards dari *Cambridge University* mengembangkan bahasa *BPCL (Basic Combined Programming Language)*. Nantinya, *BPCL* merupakan dasar bagi pengembangan bahasa B dan C yang dilakukan di laboratorium Bell.

Pascal mulai dikembangkan tahun 1968 oleh Niklaus Wirth namun baru dirilis tahun 1971. Niklaus Wirth tidak pernah menyangka bahwa bahasa pemrograman ciptaannya akan menjadi begitu populer, bahkan menjadi basis pengembangan beberapa bahasa pemrograman yang lain yang lahir setelahnya. Pada era tahun 1970 dan 1980, *Pascal* digunakan secara luas, terutama dengan hadirnya *IBM PC* dan *Macintosh*. *Compiler Pascal* tersedia baik dalam versi

⁵ Richard I. Wexelblat, *History of Programming Languages*, (New York: Academi Prees, 1981),40.

proprietary maupun *freeware*. Salah satu *compiler Pascal* yang terkenal adalah *Turbo Pascal* yang dikembangkan oleh Borland.

Bahasa pemrograman *Forth*, *Smalltalk*, dan *Icon* dirilis pada tahun yang sama pada tahun 1970. *Forth* diciptakan oleh Charles Moore didesain sebagai bahasa yang sangat simpel dan efisien. *Smalltalk* dikembangkan oleh *Xerox PARC*, pengembangannya dipimpin oleh Alan Kay. *Smalltalk* boleh dikatakan bahasa pertama yang murni *OOP (Object Oriented Programming)* atau Pemrograman Berorientasi Objek. Seluruh data yang diolah selalu dikapsulasi ke dalam objek. Beberapa versi *Smalltalk* adalah *Smalltalk-72*, *Smalltalk-74*, *Smalltalk-76*, *Smalltalk-80*, dan yang ini yang unik *GNU- Smalltalk*. *Icon* adalah salah satu turunan *Snobol*. *Icon* aslinya didesain untuk berjalan diatas sistem operasi *UNIX*, *Ms-DOS*, *Windows 32 bit*), *OS 2*, dan *Macintosh*. Pada awal pengembangan *UNIX*, berbagai macam *Shell*, dan *Bourne-Again Shell (Bash)*. Untuk keperluan pemrograman di lingkungan *Shell* tersebut terutama

Bash, diciptakan bahasa *skrip Sh. Shell Bash* bersama dengan *skrip Sh* dikembangkan oleh Steve Bourne pada tahun 1971.⁶ Bahasa pemrograman *Prolog* dirilis pada tahun 1972 dan dikembangkan oleh Alain Colmerauer dan Philip Roussel. *Prolog* dapat melakukan hal – hal yang hebat hanya dengan jumlah baris yang sedikit. Pada tahun 1972 pula dirilis bahasa pemrograman *C*. Seperti telah disebutkan sebelumnya, bahasa pemrograman *C* adalah pengganti bahasa *B*.

⁶ Abrahams Paul, “History Of Programming”, American Mathematical Society, <http://www.jstor.org/stable/2004521>

Pengembangnya adalah Dennis M. Ritchie pada saat bekerja di laboratorium Bell, Murray Hill, New Jersey.

Pada tahun 1975 dirilis tiga bahasa pemrograman yang sama sekali tidak populer, yaitu *Tiny Basic*⁷, *Scheme*, dan *RatFOR*. *Tiny Basic* dirilis oleh Bob Albrecht dan Dennis Alison namun diimplementasikan oleh Dick Whipple dan John Arnold. Dari namanya jelas terlihat bahwa *Tiny Basic* menggunakan dasar bahasa *Basic*. Kata *Tiny* ditambahkan karena bahasa tersebut digunakan untuk mikrokomputer yang RAM nya hanya 2 KB. *Scheme* merupakan dialek *LISP* dikembangkan di *MIT*. Berbeda dari *LISP*, *Scheme* memiliki banyak tipe data, diantaranya adalah simbol, beberapa tipe numeris, karakter dan *string*, *list*, *vektor*, *bit string*, *record*, *association list*, tabel hash.

Bahasa pemrograman *RatFor* merupakan singkatan dari *Rational FORTRAN* yang dikembangkan oleh B.W. Keminhan. *RatFOR* pada dasarnya adalah dikembangkan *FORTRAN*, namun mengizinkan penggunaan kontrol struktur seperti bahasa C. Pada tahun itu juga 1975, Bill Gates dan Paul Allen merilis bahasa *Basic* versi mereka sendiri dan memulai sebuah usaha dalam bidang perangkat lunak dengan mendirikan perusahaan Microsoft. *Awk* merupakan bahasa pemroses String yang dikembangkan di laboratorium Bell. Banyak digunakan di lingkungan *UNIX* untuk transformasi data dan program parsing salah satu varian *awk* yang paling bertenaga adalah *Gawk* (*GNU Awk*).

⁷ Sachinparashar sharma, "Java Programming Language" Academia.edu, https://www.academia.edu/10564468/JAVA_Programming_Language

Di akhir era tahun 1970 *IBM* banyak memproduksi komputer mainframe. Untuk memenuhi kebutuhan pemrograman di lingkungan mainframe, *IBM* menciptakan bahasa pemrograman *Rexx* yang kemudian berkembang cukup populer. Dilingkungan *Windows*,⁸ *Rexx* berani bersaing dengan *Visual Basic*, *Java*, *Perl*, dan *TCL*. Dilingkungan *UNIX*, *Rexx* bersaing dengan *Perl*, *Tcl*, *Java*, dan pemrograman *Shell*.

Dilingkungan *IBM OS 2*, bahasa pemrograman *Rexx* boleh dibilang berkibar sendiri. Kendati dikembangkan oleh Niklaus Wirth (Pengembang Bahasa Pemrograman *Pascal*) sejak tahun 1977, *Modula 2* baru dirilis pada tahun 1980. Karena dikembangkan oleh orang yang sama, tidak heran jika *Modula 2* dan *Pascal* memiliki kemiripan satu sama lain. Kelebihan *Modula 2* dibandingkan *Pascal* adalah dukungan terhadap *concurrency* dan manajemen memori dinamis yang lebih baik. Sayangnya *Modula 2* tidak mendukung *inheritance*, sesuatu yang wajib didunia *OOP* yang saat itu sedang hangat-hangatnya, akibatnya usia *Modula 2* tidak panjang.

Bahasa pemrograman *MUMPS* merupakan singkatan dari *Massachusetts General Hospital Utility Multi Programming System*. Pada tahun 1977, standar *ANSI* dari *MUMPS* dirilis. Ketika *MUMPS* mulai populer di akhir era 1970 tersebut, banyak orang menyebutnya M saja. Selain lebih keren dan juga untuk menghindari konotasi negatif. M dikembangkan terutama untuk pengamanan basis data (*database*).⁹ Dari kepanjangannya, dapat diduga bahwa M lebih banyak digunakan

⁸ Richard I. Wexelblat, *History of Programming Languages*, (New York: Academi Prees, 1981),25.

⁹ Richard I. Wexelblat, *History of Programming Languages*, (New York: Academi Prees, 1981),28.

untuk sistem informasi medis. Bahasa Pemrograman *Ada* didesain khusus untuk Departemen Pertahanan AS, dirilis pada tahun 1983. Nama *Ada* diambil dari *Ada Augusta King*, programmer pertama di dunia. Bahasa pemrograman *Ada* memiliki struktur blok dan memiliki fitur-fitur yang mendukung *OOP*, namun karena tiadanya kompiler yang bersifat free (seperti yang cukup banyak tersedia untuk C++ dan Java).

Masih belum puas dengan mengembangkan *Pascal* dan *Modula 2*, Niklaus Wirth mengembangkan lagi bahasa pemrograman yang diberi nama *Oberon*. *Oberon* dirancang sedemikian rupa sehingga kompak. Kompiler pustaka dan lingkungan operasi (IDE) juga tersedia dalam 7 buah *disket floppy*. *Oberon* memiliki beberapa versi, yaitu *Oberon V4 (System 2)*, *Oberon System 3*, dan *Oberon F*. Sayangnya *Oberon* juga tidak mampu menandingi ketenaran “kakeknya” yaitu Pascal. Sekitar tahun 1980, bahasa pemrograman berorientasi objek sedang naik daun.

Bahasa C yang saat itu termasuk bahasa yang sangat populer pun akhirnya dipoles sedemikian rupa sehingga mendukung *OOP* dan diberi nama C++. Nam C++ sendiri adalah ide dari Rick Mascitti. Pertama kali diterapkan pada tahun 1983, C++ akhirnya distandarisasi dengan standar ISO dan ANSI pada november 1997.

Bahasa pemrograman *Perl (Practical Extraction and Reporting Language)* tadinya merupakan bahasa pemrograman yang dikembangkan di lingkungan sistem operasi Unix, namun kini juga berkembang untuk sistem operasi Windows. Kelemahan Perl adalah tidak adanya standarisasi dalam penggunaan bahasa pemrograman. Namun, umumnya para programer Perl merujuk pada buku

programming Pearl, and editor, karya Larry Wall, 1987, dalam membangun aplikasi dengan *Perl*¹⁰. *Perl* sendiri dikembangkan dengan menggunakan bahasa *C*, sehingga sangat portabel. *Perl* sendiri bersifat free dan tutorial mengenai penggunaannya sangat mudah dijumpai.

Pada tahun 1990, Kenneth Iverson dan Roger Hui memperkenalkan bahasa *J* pada konferensi *APL* tahun 1990. Bahasa pemrograman *J* merupakan bahasa pemrograman matematis atau fungsional yang sangat mirip dengan *APL*. *Java* dikembangkan riset di *Sun Microsystem*. bahasa pemrograman *Java* boleh dibilang mirip dengan *C++*. *Java* didesain untuk menjadi simpel dan portabel. Lebih lanjut, menurut para pengembangnya, *Java* didesain menjadi bahasa yang berorientasi objek, kuat, aman, tidak bergantung arsitektur prosesor tertentu, *portabel*, *performa* tinggi, dan dinamis.

fitur *java*. Mereka juga dikenal sebagai kata kunci *java*. Fitur *Java* berikut contoh fitur *java* :

1. Sederhana

bahasa *Java* itu sederhana karena, Sintaks didasarkan pada *C++* (sehingga lebih mudah bagi pemrogram untuk mempelajarinya setelah *C++*). menghapus banyak fitur yang membingungkan atau jarang digunakan misalnya, petunjuk eksplisit, kelebihan beban operator. Tidak perlu menghapus objek yang tidak direferensikan karena ada Pengumpulan Sampah Otomatis di *java*.

¹⁰ E Horowitz, *The Evolution of Programming Languages*, (New York: Computer Science, 1983), 54.

2. Berorientasi Objek

OOP adalah singkatan dari Pemrograman Berorientasi Objek. Ini adalah teknik yang digunakan untuk mengembangkan program yang berputar di sekitar entitas dunia nyata. Dalam model pemrograman *OOP*, program dikembangkan di sekitar data daripada tindakan dan logika. Dalam *OOP*, setiap objek kehidupan nyata memiliki properti dan perilaku. Yang dicapai melalui pembuatan kelas dan objek. Mereka berisi properti (variabel dari beberapa jenis) dan (metode). *OOP* memberikan fleksibilitas dan kompatibilitas yang lebih baik untuk mengembangkan aplikasi besar.

3. Platform independen

Platform adalah lingkungan perangkat keras atau perangkat lunak tempat program berjalan. Ada dua jenis platform berbasis perangkat lunak dan berbasis perangkat keras. Java menyediakan platform berbasis perangkat lunak. Platform Java berbeda dari sebagian besar platform lain dalam arti bahwa platform berbasis perangkat lunak yang berjalan di atas platform berbasis perangkat keras lainnya. Istilah Platform digunakan untuk lingkungan Perangkat Keras dan Perangkat Lunak yang benar-benar menjalankan perangkat lunak. Saat ini ada banyak platform yang tersedia untuk pengguna seperti *Windows*, *Linux*, *Solaris*, *Mac*. Program *Java* dapat dijalankan di semua platform ini.

Java Platform (JVM) adalah platform khusus perangkat lunak yang berjalan di semua *OS (Windows, Linux, Solaris, Mac OS)*. Ada komponen-komponen berikut dari Platform Java:

- a. **Mesin Virtual Java:** Mesin Virtual *Java* berisi kompiler untuk mengkompilasi *Bytecode (java)* ke dalam instruksi khusus mesin. *JVM* mengkompilasi *bytecode* ke dalam kode khusus mesin dan kemudian dijalankan di platform.
- b. **Antarmuka Pemrograman Aplikasi Java (API):** Di *Java* ada banyak komponen siap pakai yang dapat digunakan untuk mengembangkan aplikasi. Semua komponen ini umumnya dikenal sebagai perpustakaan di *Java*.

Karena *Java* adalah bahasa pemrograman independen platform, ia dapat berjalan di platform apa pun. Setelah file *Java* dikompilasi menjadi file *.class*, file tersebut dapat didistribusikan di platform apa pun¹¹.

4. Aman

Kemauan *Java* dijamin dikarenakan:

1. Tidak ada penunjuk eksplisit
2. Program berjalan di dalam kotak mesin virtual.

5. Kuat

Java menggunakan manajemen memori yang kuat. Ada kekurangan petunjuk yang menghindari masalah keamanan. Ada pengumpulan sampah otomatis di *java*. Ada mekanisme exception handling dan type checking di *java*. Semua poin ini membuat *java* kuat.

6. Arsitektur netral

Tidak ada fitur yang bergantung pada implementasi, misalnya ukuran tipe primitif ditetapkan.

¹¹ Sachinparashar sharma , “Java Programming Language” Academia.edu, https://www.academia.edu/10564468/JAVA_Programming_Language

7. Portabel

Kita bisa membawa bytecode java ke platform manapun.

8. IC dinamis

Java lebih cepat daripada interpretasi tradisional karena kode *byte* dengan kode asli masih sedikit lebih lambat daripada bahasa yang dikompilasi (misalnya, *C++*). Kita dapat membuat aplikasi terdistribusi di *java*. *RMI* dan *EJB* digunakan untuk membuat aplikasi terdistribusi. Kita dapat mengakses file dengan memanggil metode dari mesin manapun di internet.

Demi keamanan, para perancang *Java* tidak memasukkan fitur *pointers*, inheritance ganda, operator, *overloading*, dan *mikroprosesor*. Pertama sekali dikembangkan, *Java* bernama *Oak*¹². Namun para pengembangnya, mersa nama *Oak* tersebut kurang cocok untuk mewakili sebuah bahasa pemrograman yang sedemikian tangguh. Mereka mengadakan *brainstorming* disebut kedai kopi. Dan mereka memesan kopi jawa. Karena terkesan dengan rasa kopi tersebut, maka nama *Java* pun pun diambil sebagai nama bahasa pemrograman yang mereka kembangkan.

Ada banyak versi java yang telah dirilis. Rilis Java yang stabil saat ini adalah Java

SE 8. UNIVERSITAS ISLAM NEGERI

1. *JDK* Alpha dan Beta (1995)

2. *JDK* 1.0 (23 Januari 1996)

3. *JDK* 1.1 (19 Februari 1997)

¹² Jean e. Sammet, Programming Languages:History and fundamentals (New York: prentice-hall, inc, 1996), 25.

4. *J2SE 1.2* (8 Desember 1998)
5. *J2SE 1.3* (8 Mei 2000)
6. *J2SE 1.4* (6 Februari 2002)
7. *J2SE 5.0* (30 September 2004)
8. *Java SE 6* (11 Desember 2006)
9. *Java SE 7* (28 Juli 2011)
10. *Java SE 8* (18 Maret 2014)
11. Edisi Standar Java 8 Pembaruan 25 (14 Oktober 2014)

Bahasa pemrograman *Python*¹³, *Python* adalah bahasa scripting tingkat tinggi, ditafsirkan, interaktif dan berorientasi objek. *Python* dirancang agar sangat mudah dibaca. Ini sering menggunakan kata kunci bahasa Inggris sedangkan bahasa lain menggunakan tanda baca, dan memiliki lebih sedikit konstruksi sintaksis dari pada bahasa lain.

- a. *Python* Ditafsirkan: *Python* diproses saat runtime oleh interpreter. Anda tidak perlu mengkompilasi program Anda sebelum menjalankannya. Ini mirip dengan *PERL* dan *PHP*.
- b. *Python* bersifat Interaktif: Anda sebenarnya dapat duduk di prompt *Python* dan berinteraksi dengan juru bahasa secara langsung untuk menulis program Anda.
- c. *Python* Berorientasi Objek: *Python* mendukung gaya Berorientasi Objek atau teknik pemrograman yang merangkum kode di dalam objek.

¹³ Richard I. Wexelblat, *History of Programming Languages 2*, (New York: Academi Prees, 1981),28

- d. Python adalah Bahasa Pemula: Python adalah bahasa yang bagus untuk pemrogram tingkat pemula dan mendukung pengembangan berbagai aplikasi mulai dari pemrosesan teks sederhana hingga browser WWW hingga game.

Python berasal dari banyak bahasa lain, termasuk *Modula-3*, *C*, *C++*, *Algol-68*, *SmallTalk*, *Unix shell*. Pada saat beliau mulai mengimplementasikan *Python*, Guido van Rossum juga membaca skrip yang diterbitkan dari "*Monty Python's Flying Circus*"¹⁴.

Python sekarang dikelola oleh tim pengembangan inti di institut tersebut, meskipun Guido van Rossum masih memegang peran penting dalam mengarahkan kemajuannya berikut contoh perkembangan bahasa pemrograman python:

1. Python 1.0 dirilis pada 20 Februari 1991.
2. Python 2.0 dirilis pada 16 Oktober 2000 dan memiliki banyak fitur utama baru, termasuk siklus pendeteksian pengumpul data base dan dukungan untuk Unicode. Dengan rilis ini, proses pengembangan diubah dan menjadi lebih transparan dan didukung komunitas.
3. Python 3.0 (yang pada awal perkembangannya biasanya disebut sebagai Python 3000), rilis utama yang tidak kompatibel dengan versi sebelumnya, dirilis pada 3 Desember 2008 setelah periode pengujian yang lama. Banyak fitur utamanya telah di-porting kembali ke seri versi Python 2.6.x dan 2.7.x yang kompatibel mundur.

¹⁴ E Horowitz, *The Evolution of Programming Languages*, (New York: Computer Science, 1983), 54.

4. Pada bulan Januari 2017, Google mengumumkan pengerjaan transkompiler Python 2.7 to go, yang menurut perkiraan The Register merupakan tanggapan atas akhir masa pakai Python 2.7 yang direncanakan.

Python pertama mengkompilasi kode sumber Anda (pernyataan dalam file Anda) ke dalam format yang dikenal sebagai kode byte. Kompilasi hanyalah langkah terjemahan, dan kode byte adalah tingkat yang lebih rendah, representasi independen platform dari kode sumber Anda. Secara kasar, *Python* menerjemahkan setiap pernyataan sumber Anda menjadi sekelompok instruksi kode byte dengan menguraikannya menjadi langkah-langkah individual. Terjemahan kode byte ini dilakukan untuk mempercepat eksekusi kode byte dapat dijalankan jauh lebih cepat daripada pernyataan kode sumber asli dalam file teks Anda.

Aplikasi Python:

- a. Pemrograman Sistem
- b. GUI
- c. Internet Scripting
- d. Integrasi Komponen
- e. Pemrograman Basis Data
- f. Rapid Prototyping
- g. Pemrograman Numerik dan Ilmiah

Mesin Virtual Python:

Setelah program Anda dikompilasi menjadi kode *byte* (atau kode byte telah dimuat dari pyc), program tersebut dikirim untuk dieksekusi ke sesuatu yang umumnya dikenal sebagai mesin virtual *python* (PVM).

dimulai dengan rilis *Visual Basic* pada tahun 1991. Bahasa *Basic* yang digunakan tidak beda dengan dialek-dialek *Basic* yang lain, namun dibarengi dengan desain pemrograman secara *visual*. Salah satu yang mendukung kepopuleran *Visual Basic* adalah kemampuan untuk diperluas (*extensibility*), yang berarti ada komponen-komponen yang dapat disematkan ke dalam *Visual Basic*.¹⁵

Begitu banyak komunitas programmer yang mendesain kontrol-kontrol ekstensi tersebut dalam bentuk *Visual Basic Control (VBXs)* dan *activeX Controls*. *Ruby* yang jelas bukan nama maskot kartu grafis *ATI* yang seksi itu, tetapi nama bahasa pemrograman. *Ruby* sangat mirip dengan *Pearl*, bedanya adalah pada cara pengolahan data.

Pearl memperlakukan *Ruby* mengimplementasikan semuanya sebagai objek. Seperti disebutkan sebelumnya, salah satu *compiler Pascal* yang terkenal adalah produksi *Borland*, yaitu *Turbo Pascal*. Karena terkesan dengan *Visual Basic*,¹⁶ maka *Borland* juga ingin memproduksi *Visual Pascal*. Maka pada tahun 1995, lahirnya *Delphi*, sebuah lingkungan pemrograman bersifat visual yang memiliki dasar bahasa *Pascal*. *Delphi* bahkan sempat menjadi bahasa pemrograman favorit para programmer pada akhir dekade 1990 sebelum popularitasnya menurun drastis pada tahun 2000 hingga sekarang.

Bahasa pemrograman *Dylan (Dynamic Language)* dikembangkan oleh *Apple Computer* pada tahun 1992. Sintaks *Dylan* merupakan ide yang orisinal, namun secara garis besar, strukturnya mirip dengan bahasa *Scheme*. W. Walaupun

¹⁵ Jean e. Sammet, *Programming Languages: History and fundamentals* (New York: prentice-hall, inc, 1996), 10.

¹⁶ E Horowitz, *The Evolution of Programming Languages*, (New York: Computer Science, 1983), 80

ditujukan untuk platform Macintosh, namun belakangan juga dapat digunakan diatas platform Unix dan Windows.

Pada tahun 1994, Netscape merilis sebuah bahasa skrip Bernama *live Script*. *Netscape* menginginkan adanya “perilaku” yang dinamis pada dunia web. Sun *Microsystem*, penghembangan *Java*, tertarik dengan kemampuan *Live Script*. Sun dan *Netscape* kemudian bekerjasama dengan mengembangkan *Live Script* menjadi lebih hebat lagi dan diberi nama *Java Script*. *Java Script* sangat mirip dengan *Java*, namun lebih sederhana dan digunakan khusus pada web browser, terutama browser keluaran *Netscape*. *Internet Explorer* produk *Microsoft* juga mendukung *Java Script*, namun dengan bahasa *script*, mereka memodifikasi sendiri dan diberi nama *Jscript*.

Tidak mau kalah dengan *Netscape*, *Microsoft* juga merilis sendiri sebuah bahasa script untuk web browser. Bahasa tersebut adalah *VB Script*. Walaupun dikembangkan sejak 1995, *VBS cript* baru dirilis bersama dengan internet Explorer 3.0 pada tahun 1997. *VBS cript* atau *Visual Basic Scripting Edition*, bahasa yang digunakannya adalah bahasa Basic dan ditujukan untuk pemrograman web. Satu-satunya browser yang mendukung *VBS cript* adalah *Internet Explorer*.¹⁷

Pada pertengahan dekade tahun 1990 mulai dirasakan keterbatasan bahasa *script* yang didukung oleh browser. Penyebabnya adalah pengelolaan dilakukan oleh klien, bukan server. Maka dikembangkanlah bahasa *script* yang diolah oleh *server*. Sejak itu mulailah bermunculan *server side scripting*. *ASP (Active Server*

¹⁷ “The History of Programming Languages” *croshero*, [404643b570eba1c8ccc4833a24ee358f5303f4ba.pdf](#).

Pages) adalah ciptaan *Microsoft*, merupakan teknologi bahasa script di lingkungan server yang dapat digunakan untuk membangun halaman web yang interaktif dan membangun aplikasi web yang *powerfull*. Karena ASP lebih condong ke arah teknologinya, bukan bahasanya, maka ada beberapa bahasa yang dapat digunakan. Bahasa *script ASP* yang umum digunakan adalah *VB Script* dan *Jscript*. Untuk dapat menjalankan *ASP*, diperlukan *IIS (Internet Information Server)*.

PHP merupakan singkatan dari rekursif dari *PHP (Hypertext Processor)*. Ada juga yang menyebutnya *Personal Home Page tools*. akan tetapi kepanjangan yang digunakan adalah yang pertama. Berbeda dengan *ASP*, *PHP* bukan merujuk pada lingkungan server side scripting, namun pada bahasa script itu sendiri. Bahasa *script PHP* menggunakan dialek bahasa *C*.

Sejauh ini bahasa *C#* (bahasa *C sharp*) adalah bahasa pemrograman termuda. *C#* dikembangkan oleh *Microsoft* dan pertama sekali diajukan pada *ECMA Standar Group* sekitar pertengahan 2000. *C#* didesain untuk memiliki ketangguhan yang sama dengan *C++* atau *Java*, namun semudah *Visual Basic*. Dengan mengetahui tentang sejarah pemrograman, diharapkan dapat memberi motivasi dan makin tertarik dengannya untuk menjadi programmer yang baik, bahkan dapat menciptakan bahasa pemrograman yang baru

B. Konferensi HOPL

Perkembangan bahasa pemrograman yang pesat ini menarik perhatian lembaga *ACM Special Interest Group On Programming Languages (SIGPLAN)* dalam mengadakan konferensi untuk mendokumentasikan sejarah bahasa pemrograman hingga titik waktu tersebut, yang dinamakan *Conference on the*

History of Programming Languages (HOPL).¹⁸ Konferensi dilaksanakan sebanyak dua kali. Konferensi pertama (*HOPL*) pada tahun 1978 memilih tiga belas bahasa pemrograman yang memenuhi kriteria sebagai berikut:

1. Telah digunakan selama setidaknya sepuluh tahun.
2. Memiliki pengaruh yang signifikan terhadap bidang programming.
3. Masih digunakan hingga saat konferensi diadakan.

Dari ketiga kriteria tersebut, bahasa-bahasa pemrograman yang terpilih pada seleksi pertama adalah *ALGOL*, *APL*, *APT*, *BASIC*, *COBOL*, *FORTRAN*, *GPSS*, *JOSS*, *JOVIAL*, *LISP*, *PL 1*, *SIMULA*, *SNOBOL*.

Kemudian konferensi pada kedua (*HOPL-II*) diadakan pada tahun 1993 untuk mematangkan konsep sejarah bahasa pemrograman yang bersifat kronologis. Terdapat perubahan yang cukup signifikan terhadap bahasa-bahasa pemrograman yang terpilih pada konferensi ini, mengingat celah waktu antara konferensi yang pertama dengan yang kedua cukup jauh, dan bahasa pemrograman terus mengalami perkembangan selama jangka waktu tersebut. Bahasa-bahasa pemrograman yang terpilih pada seleksi kedua adalah *ADA*, *ALGOL 68*, *C*, *C++*, *CLU*, *DSL*, *FORMAC*, *FORTH*, *ICON*, *LISP*, *PASCAL*, *PROLOG*, *SMALTAK*.

Dapat dilihat bahwa ada beberapa bahasa pemrograman yang mungkin seharusnya ada, namun tidak terdapat dalam *HOPL-II*. Misalnya, *FORTRAN* dan *COBOL* yang dipilih pada konferensi pertama, tidak terpilih dalam konferensi kedua. Hal ini dikarenakan bahasa tersebut masih mengalami perubahan yang signifikan pada

¹⁸ Jean e. Sammet, *Programming Languages: History and fundamentals* (New York: prentice hall, inc, 1996), 40

saat diadakannya konvrensi HOPL II. Misalnya, bahasa FORTRAN baru disempurnakan pada tahun 1990 dengan dirilisnya versi *FORTRAN 90*, sedangkan bahasa yang dipilih adalah bahasa yang telah relatif stabil dan telah digunakan sepuluh tahun sebelum konferensi kedua diadakan¹⁹.

Berikut ini deskripsi dari bahasa pemrograman yang dipresentasikan dalam konferensi kedua (HOPL-II) :

1. *ALGOL 68* Bahasa ini merupakan kelanjutan dari *ALGOL 60*, yang sebelumnya menjadi bahasa pertama yang tidak bergantung pada mesin (*machine independent language*). Bahasa ini merupakan bahasa yang sangat berpengaruh terhadap bahasa pemrograman berikutnya, terutama Pascal, C, dan Ada (yang secara tidak langsung berpengaruh pula terhadap bahasa-bahasa modern yang dipengaruhi oleh ketiga bahasa tersebut). Kontribusi dari bahasa pemrograman ini diantaranya adalah diterapkannya user-defined data structures, tipe referensi, dan array dinamik (flex array). Bahasa ini dibangun dengan harapan bidang penggunaannya akan lebih luas dengan banyaknya sintaks dan semantik yang disediakan. Namun Sayangnya hal ini membuat ALGOL 68 lebih sulit untuk digunakan dibanding dengan versi sebelumnya, ALGOL 60, sehingga kalah bersaing dengan C dan Pascal yang relatif lebih umum digunakan karena pemakaiannya lebih mudah.²⁰
2. Pascal Bahasa ini dibangun pada tahun 1968-1969 oleh Niklaus Wirth yang keluar dari tim ALGOL 68. Pascal Pada masanya merupakan bahasa prosedural

¹⁹ Linda W. Friedman, “sejarah singkat bahasa pemrograman” *compulang* 1991.

²⁰ Nobel Noah, “History and Evolution of Computer Programming Languages”, *Steemit*. 2017. https://steemit.com.translate.googleusercontent.com/@noble-noah/history-and-evolution-of-computer-programming-languages?x_tr_sl=en&x_tr_tl=id&x_tr_hl=id&x_tr_pto=scv

yang sangat berpengaruh. Bahkan, Pascal masih digunakan dalam latihan pemrograman struktural dan struktur data hingga sekarang.

3. *Concurrent Pascal* Bahasa ini dibangun oleh *Per Brinch Hansen* secara terpisah dari *Pascal* Yang dibuat oleh Niklaus Wirth. Perbedaan yang signifikan dari *Concurrent Pascal* (atau sering disebut *PASCAL-FC*) adalah sistemnya yang bersifat konkuren, berbeda dengan Pascal yang bersifat sekuensial. Hal ini memungkinkan komputer untuk mengeksekusi beberapa perintah sekaligus. Bahasa ini ditujukan untuk diaplikasikan dalam sistem operasi ataupun sistem memonitor secara real-time dalam beberapa komputer yang bersifat shared memory Pada dasarnya sintaks dan semantik yang digunakan sama dengan Pascal yang bersifat sekuensial, hanya saja fitur-fitur seperti variant records, goto statement, procedures as parameters, pointer types, packed arrays, dan file types, dalam Pascal Yang bersifat sekuensial ditiadakan untuk membuat bahasa ini lebih sederhana.
4. Ada bahasa pemrograman ini sebenarnya dibangun oleh tim yang dipimpin Jean Ichbiah Untuk *United States Department of Defense (DoD)* pada tahun 1977 hingga 1983 untuk menggantikan ratusan bahasa pemrograman yang sebelumnya digunakan oleh *United States Department of Defense (DoD)* ²¹. Nama bahasa pemrograman yang dipengaruhi oleh sintaks *ALGOL 68*. ini diambil dari orang yang dianggap sebagai pemrogram komputer pertama, Ada Lovelace dan Francis Pengembangan bahasa ini melibatkan ratusan orang, biaya besar, dan waktu pengembanganyang relatif lama. *Compiler* yang pertama

²¹ Linda W. Friedman, “sejarah singkat bahasa pemrograman”, compulang 1991.

dibuat pada tahun 1977 sangatlah rumit. *Compiler* yang dapat digunakan secara luas baru dirilis 5 tahun kemudian setelah desain bahasanya telah disempurnakan. Pada awalnya bahasa Ada difokuskan untuk diaplikasikan pada sistem embedded and real-time. Namun seiring perkembangannya Ada sekarang bisa digunakan untuk pemrograman berorientasi objek. Fitur yang pekat dalam bahasa ini adalah *strong typing*, mekanisme modular menggunakan *package exception handling dan runtime checking*.

5. Lisp Bahasa ini dibangun oleh John McCarthy pada tahun 1958 dan desainnya dipublikasikan pada tahun 1960, menjadikannya salah satu bahasa pemrograman tertua. Beberapa sumber mengatakan bahwa Lisp merupakan salah satu bahasa pemrograman terbaik karena strukturnya yang memudahkan pemrograman untuk memperluas bahasa yang digunakan ataupun mengimplementasikan dialek baru. Fitur yang dimiliki Lisp membuatnya digunakan secara luas dalam bidang penelitian kecerdasan buatan (AI). Kontribusi dari bahasa ini diantaranya penggunaan tree data structures (functional programming) automatic storage management, dynamic typing, conditionals, high-order functions, recursion, dan self-hosting compiler. Namun kontribusi yang paling dominan adalah sebagai pionir dalam penggunaan fungsi dalam pemrograman.

6. Prolog Bahasa pemrograman ini pada awalnya tidak ditujukan untuk menjadi sebuah bahasa pemrograman, melainkan untuk memproses bahasa natural. Prolog sendiri dibangun pada tahun 1972 oleh Colmerauer Bersama dengan Philippe Roussel. Seiring perkembangannya, Prolog Menjadi bahasa pemrograman pengolah logika yang berasosiasi dengan kecerdasan buatan dan

komputasi linguistik. Hingga saat ini Prolog masih digunakan dalam pembuatan sistem pakar, pembuktian teori, dan pemrosesan bahasa natural karena kemampuannya dalam pemrosesan logika. Dalam Prolog, logika program diekspresikan dengan relasi dan pengolahan *query*

7. *Discrete Simulation Language* bahasa yang termasuk dalam kategori ini adalah bahasa yang memiliki model operasi dari sistem yang sifatnya diskrit. dengan kata lain, setiap *event* dalam suatu waktu menandakan perubahan *state* dari suatu sistem, lalu simulasi hanya dilakukan pada *event* tertentu. Berbeda dengan Continuous Simulation yang mendeteksi perubahan-perubahan yang terjadi pada sistem setiap waktunya. Karena discrete simulation melakukan simulasi lebih sedikit dibanding dengan *continuous simulation*, maka discrete simulation akan bekerja lebih cepat dalam melakukan pemodelan. Contoh yang paling populer adalah bahasa *Simula*. Bahasa ini dibangun oleh Ole-Johan Dahl dan Kristen Nygaard sekitar tahun 1967.²² *Simula* Dapat dikatakan sebagai bahasa pemrograman berorientasi objek yang pertama karena fiturnya yang disebut coroutine (semacam sub program) yang diimplementasi dalam struktur yang disebut class.

8. *FORMAC* Bahasa ini dibangun oleh Jean E. Sammet bersama *IBM* pada tahun 1964.²³ *FORMAC* merupakan bahasa pertama yang membuat simbol-simbol matematika dapat dikomputasikan secara praktis dan signifikan dalam pemrograman. *FORMAC* memiliki fitur untuk melakukan komputasi,

²² Suparparto, *Bahasa Pemrograman*, (Jakarta: Departemen Pendidikan Nasional, 2018), 50

²³ Mutiara Ramadya Kusuma, " Evolution of Programming Language", *Scribid*, 2010: 6. <https://id.scribd.com/doc/65375992/Evolusi-Bahasa-Pemrograman>.

manipulasi, dan penggunaan ekspresi simbolik dalam persoalan aljabar. Sistem inilah nantinya akan diadaptasi oleh *FORTRAN IV* dan membuat bahasa tersebut menjadi lebih sempurna. *FORMAC* menjadi sebuah ekstensi dari *FORTRAN* dan memiliki tipe data tersendiri, yaitu "*FORMAC Variable*", Nama tersebut digunakan untuk membedakan operasi aljabar dengan operasi matematika biasa.

9. *CLUB* Bahasa ini dibuat oleh Barbara Liskov dan murid-muridnya di *MIT* sekitar tahun 1974. *CLU* adalah bahasa pertama yang mengimplementasikan *constructor* untuk *abstract data types (ADT)*, yang menjadi salah satu kunci dari pemrograman berorientasi objek. Namun terdapat beberapa kunci utama pemrograman berorientasi objek yang tidak terdapat dalam *CPU*, sehingga *CLU* disebut sebagai bahasa *object-based*, bukan sebagai *object-oriented*. Konsep class dari Semula Digunakan kembali dalam bahasa ini, sehingga *CLU* lebih mendekati bahasa pemrograman berorientasi objek dibanding dengan *Simula*.
10. *Smalltalk* Bahasa ini dibangun dan dikembangkan di *Xerox Palo Alto Research Center (Xerox PARC)* oleh Alan Kay, kemudian dilanjutkan oleh Adele Goldberg. *Smalltalk* merupakan bahasa pertama yang mengimplementasikan seluruh kunci dari konsep *OOP* berupa data *Abstraction*, *inheritance*, dan *dynamic type*. Disamping kelengkapan konsep *OOP*, *Smalltalk* merupakan pioneer dari konsep *What You See is What You Get (WYSIWYG)*.²⁴ Dengan kata lain, *Smalltalk* adalah bahasa pemrograman yang berhasil mengimplementasikan

²⁴ Rickiy R. Sargi., "Pemrograman Dan Bahasa Pemrograman", *Researchgate*, Juni 2018, https://www.researchgate.net/publication/329885312_PEMROGRAMAN_DAN_BAHASA_PEMROGRAMAN?enrichId=rgreq-7ae7a0e69a98a2d8f6aaa6caf166697fXXX&enrichSource=Y292Z_XJOYWdlOzMyOTg4NTMxMjtBUzo3MDcyNzU1ODAzODAxNDIAMTU0NTYzOTA0NzkwNg%3D%3D&el=1_x_2&_esc=publicationCoverPdf

Graphical User Interface (GUI). Konsep ini merupakan pintu menuju bahasa pemrograman generasi selanjutnya, yaitu pemrograman berbasis visual. Built-in debugging dan object inspection tools yang dimiliki oleh environment Smalltalk menjadi standar dari Integrated Development Environment yang ada saat ini.

11. Icon Bahasa ini dibangun oleh Ralph Griswold pada tahun 1977. Fitur Utama dari Icon adalah pengolahan string dan pola tekstual. Icon adalah hasil pengembangan dari bahasa sebelumnya, yaitu SNOBOL yang juga merupakan bahasa pengolah string.²⁵ Pada awalnya Icon tidak bersifat object-oriented, namun pada perkembangannya dibuatlah ekstensi yang mendukung sifat tersebut pada tahun 1996, yang dinamakan Idol, sehingga nantinya Icon berubah nama menjadi Unicon. Perbedaan yang cukup signifikan antara Icon dengan bahasa lain pada masanya adalah penggunaan konsep goal-directed execution, yaitu pengembalian ekspresi success or failure dalam operasi logika, dibanding mengembalikan nilai boolean seperti bahasa umumnya. Icon memiliki fitur string scanning untuk mempermudah dalam membandingkan string dengan menghindari hal-hal mendetail yang tidak perlu dalam menganalisis string.
12. Forth Bahasa ini dibangun oleh Charles H. Moore pada sekitar tahun 1970. Berbeda dengan bahasa pemrograman berskala besar lainnya yang pengembangannya dibantu oleh sebuah perusahaan besar tertentu, pengembangan Forth hanya dilakukan oleh satu orang, yaitu membuatnya sendiri. Namun dalam perkembangannya, bahasa ini menarik perhatian

²⁵ Jean e. Sammet, Programming Languages: History and fundamentals (New York: prentice-hall, inc, 1996), 40.

pemrogram dalam menciptakan tools untuk menyelesaikan masalah dalam sebuah aplikasi. Forth bersifat stack-based, yaitu sistem yang bergantung kepada model stack machine dalam melakukan passing parameter .Forth yang sifatnya portable menjadi populer pada tahun 1980 karena cocok digunakan dalam mikrokomputer pada masanya.²⁶

13. C Bahasa ini dibangun oleh Dennis Ritchie dari AT&T Bell Labs pada tahun 1972. Bahasa C khusus dibuat untuk pemrograman secara umum dan menjadi bahasa yang paling luas digunakan di seluruh dunia. *Compiler C* dapat digunakan hampir di seluruh arsitektur komputer dan sistem operasi. Desain Bahasa C menyediakan kompiler yang memberikan akses level rendah ke memori sehingga hasil kompilasi relatif lebih efisien untuk dipindahkan ke dalam bahasa mesin. Kemampuan bahasa C tersebut memungkinkan terciptanya sistem operasi baru seperti UNIX. Bahkan *Microsoft* kemudian mengadopsi bahasa C untuk membuat sistem operasi Windows pada tahun 1980.
14. C++ bahasa ini dibangun oleh Bjarne Stroustrup di *AT&T Bell Labs* pada tahun 1985. Pada dasarnya, C++ bisa dikatakan sebagai pengembangan dari bahasa C yang diberi fitur *OOP* yang terdapat dalam *SIMULA*. Akibatnya, C++ menjadi bahasa yang besar dan kompleks karena mendukung baik procedural programming maupun object oriented programming.

Latar Belakang dibangunnya bahasa ini adalah menciptakan program berorientasi objek yang dapat digunakan dalam pembuatan aplikasi berskala besar,

²⁶ Jean e. Sammet, *Programming Languages: History and fundamentals* (New York: prentice-hall, inc, 1996), 40.

berbeda dengan bahasa SIMULA yang hanya mampu mendukung pembuatan aplikasi berbasis OOP berskala kecil. Disamping itu, C++didesain agar tidak hanya mudah digunakan seperti SEMULA, tetapi juga dapat mengungguli bahasa Dalam ketepatan koreksi logika program, run-time speed dan kompresi ukuran kode yang dihasilkan.²⁷



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

²⁷ Perov Maxim, “Sejarah Perkembangan Bahasa Pemrograman”, (Tesis, Universitas Moskow, 2016), 35.

BAB IV

DAMPAK PERKEMBANGAN BAHASA PEMROGRAMAN

A. Dampak Perkembangan Bahasa Pemrograman Tahun 1955 - 1995

Tahun 1650 sampai dengan 1955, disebut sebagai era industri. Era ini dimulai dengan terjadinya revolusi industri, yaitu sejak ditemukannya mesin-mesin industri. Tenaga kerja manusia di dalam pabrik mulai diganti dengan mesin. Era informasi dimulai dengan ditemukannya komputer pada tahun 1955. Pada era ini teknologi informasi sudah menggunakan komputer dan pemrosesan informasi menjadi lebih cepat, pemrosesan dan penyimpanan informasi menjadi lebih murah, dan tidak banyak memakan tempat dan waktu.

Di bidang akuntansi, sistem pemrosesan informasi akuntansi berbasis komputer banyak ditawarkan dengan tujuan untuk memberikan kemudahan bagi para akuntan untuk menghasilkan informasi yang dapat dipercaya, relevan, tepat waktu, lengkap, dapat dipahami, dan teruji.

Dalam era bisnis global, pengaruh kemajuan teknologi informasi tidak dapat dihindarkan lagi, seperti penggunaan telepon, faksimili, komputer, dan satelit dalam berbagai aktivitas sarana berkomunikasi perusahaan. Teknologi informasi memungkinkan manusia untuk memperoleh informasi dari tempat yang berjauhan dalam waktu yang singkat dan dengan biaya yang murah.

Perkembangan Ilmu Pengetahuan dan Teknologi saat ini semakin hari semakin berkembang pesat dalam hal fungsinya yang semakin bermanfaat, dalam bentuk yang semakin bagus dan penggunaan yang semakin mudah

digunakan oleh masyarakat luas. Hal ini yang membuat banyak masyarakat sangat antusias dalam hal teknologi.

Teknologi sama halnya dengan sebuah gunting, jika digunakan pada kertas dengan cara-cara baik maka akan menghasilkan sebuah potongan atau karya yang indah pada kertas itu yang pastinya akan bermanfaat, dan sebaliknya jika kita menggunakan gunting itu untuk hal-hal negatif, maka itu pun dapat melakukan hal-hal yang jelek pula, yang akan merugikan bagi diri sendiri ataupun bagi orang lain.¹

Sejak ditemukannya WWW (World Wide Web) telah mengubah pola komunikasi di dunia. Karena informasi dari seluruh dunia tersedia secara luas dan bebas di internet. Hal ini membuat semua informasi mudah diakses di mana saja dan kapan saja. Semua berkat teknologi modern. Teknologi modern telah menggantikan radio dengan televisi, dan sekarang bahkan televisi telah didigitalkan menjadi “LCD” dan “LED”. Berbagai upaya sedang dilakukan untuk menciptakan sumber informasi yang lebih baik.

Pada zaman yang serba maju dan modern ini kita dapat mencari tempat tertentu dan bahkan menentukan tujuan secara spesifik. Teknologi yang digunakan untuk melakukan pencarian tempat tersebut adalah Google Maps dengan memanfaatkan google maps memudahkan kita untuk mencari lokasi secara tepat. Aplikasi buatan google tersebut sangat membantu untuk melakukan navigasi pada daerah yang belum pernah kita kunjungi. Hal tersebut tentunya

¹ Deni Hardianto, “Pemanfaatan software komputer untuk Meningkatkan kecerdasan emosi anak”, (2016):7, <http://staffnew.uny.ac.id/upload/132310873/penelitian/Pemanfaatan+Software+Komputer+Untuk+Meningkatkan+Kecerdasan+Emosi.pdf>.

akan menghemat waktu kita untuk mencapai tempat yang dituju secara cepat dan tepat.

Salah satu tujuan utama teknologi adalah membuat segala sesuatu lebih murah dan lebih terjangkau bagi orang-orang. Oleh karena itu, teknologi memangkas proses bisnis yang cenderung panjang dan berbelit hal tersebut membuat biaya yang dikeluarkan oleh perusahaan maupun pengguna lebih kecil dan lebih efisien. Contohnya penggunaan kasir manusia bisa digantikan dengan kasir mesin selain itu penggunaan E-Card untuk melakukan pembayaran pada jalan tol.

Teknologi benar-benar telah menghasilkan digitalisasi dan modernisasi di berbagai bidang. Baik itu bidang kedokteran atau pertanian atau elektronik, teknologi telah menghasilkan revolusi global. Teknik yang lebih baik dalam bertani telah menghasilkan makanan yang lebih banyak dan lebih sehat. Teknik "pertanian layer" membutuhkan lebih sedikit ruang dan menghasilkan lebih banyak makanan. Kesehatan hewan yang lebih baik menjamin hasil yang lebih banyak dari produk susu dan unggas. Sektor kesehatan juga mendapat banyak manfaat dari ledakan teknologi. Bahkan penyakit yang tidak dapat disembuhkan seperti kanker memiliki obat yang tepat sekarang.

Pada Era Teknologi komunikasi menjadi sangat mudah bagi orang untuk terhubung dan berkomunikasi dengan orang-orang di seluruh dunia. Ponsel adalah contoh terbaik dari teknologi komunikasi. Melalui ponsel, orang dapat menghubungi dan berkomunikasi dengan orang, teman, dan kerabat di seluruh dunia dengan sangat mudah dan dengan biaya lebih murah. Sebelum penemuan

ponsel dan telepon, orang menggunakan surat untuk komunikasi dengan kerabat dan teman mereka. Sangat sulit bagi orang untuk berkomunikasi melalui surat karena mengirim surat membutuhkan banyak waktu dan juga biaya. Tetapi setelah penemuan telepon dan telepon genggam, menjadi sangat mudah bagi orang untuk berkomunikasi dengan kerabat mereka dalam hitungan detik. Melalui ponsel, kita juga dapat berkomunikasi dengan kerabat dan teman melalui video call. Dan saya pikir ini adalah salah satu keuntungan terbaik dari teknologi bagi manusia. Ini menghemat waktu dan juga uang bagi orang-orang dalam komunikasi.

Perbankan juga mendapatkan pengaruh yang sangat signifikan oleh pesatnya perkembangan teknologi. Sekarang teknologi digunakan di setiap bank. Tujuan utama teknologi adalah untuk mempermudah pekerjaan masyarakat dan karena teknologi perbankan menjadi sangat memudahkan masyarakat. Transaksi online dan mesin atm sangat memudahkan orang untuk melakukan transaksi apapun dari rumah mereka atau dari mesin ATM manapun. Mereka tidak perlu datang ke bank dan menunggu serta bersusah payah mengantri, mereka dapat dengan mudah melakukan transaksi apapun dari rumah mereka secara online atau dari mesin ATM di daerah mereka.

Teknologi mempunyai banyak manfaat dan pengaruh bagi masyarakat. Sekarang ini di Indonesia banyak sekali orang yang menggunakan percakapan dengan artian-artian atau bahasa-bahasa teknologi atau bahasa asing. Semua ini mempunyai dampak positif dan bisa juga mempunyai dampak negatifnya.

Berikut dampak positif kemajuan teknologi :

1. Membantu manusia beraktivitas

Bahasa pemrograman Komputer sangat membantu masyarakat, khususnya dalam menjalankan aktivitas, seperti belajar dan bekerja. Contohnya membantu pelajar dalam mengetik dan menyusun tugas. membantu karyawan dalam mengetik dan memasukkan data pekerjaan. Menyimpan dokumen atau data penting Selain memudahkan manusia, bahasa pemrograman komputer juga bermanfaat untuk menyimpan berbagai dokumen atau data penting, seperti hasil pekerjaan, tugas sekolah, foto. Perhitungan.

dalam kehidupan masyarakat, bahasa pemrograman komputer juga bermanfaat untuk melakukan perhitungan. Sebagai contoh *program Microsoft Excel* digunakan untuk menghitung data berupa angka.

2. Melakukan pengolahan data

Manfaat lain bahasa pemrograman komputer bagi kehidupan masyarakat adalah pengolahan data. Ada banyak program aplikasi yang bisa digunakan untuk melakukan pengolahan data. Contohnya program *Microsoft Word* untuk pengolahan data berupa *teks*, *Microsoft Excel* untuk pengolahan angka, *Microsoft Power Point* untuk pengolahan data dalam bentuk presentasi, dan masih banyak lagi. Mencari informasi di internet.

Bahasa pemrograman Komputer juga bisa dimanfaatkan untuk mencari informasi penting di internet Pencariannya bisa dilakukan di *Google Chrome*, *Opera Mini*, *Internet Explorer*, dan *Mozilla Firefox*.

3. Mencari sebuah tempat

Bahasa pemrograman komputer juga bisa dimanfaatkan untuk menemukan suatu tempat. Dengan menggunakan aplikasi google map kita bisa menemukan suatu tempat serta mengantar kita ke lokasi yang kita tuju dengan akurat dan cepat.²

4. Dalam bidang agama

Bahasa pemrograman computer juga bisa dimanfaatkan untuk beribada, seperti Ketika kita sedang berada di luar kota, kita bisa memanfaatkan aplikasi Al quran, dalam aplikasi tersebut terdapat, penunjuk waktu sholat, dan penentu arah kiblat.

Pada saat ini bahasa pemrograman yang sangat bereperan penting dalam kamajuan teknologi adalah bahasa pemrograman Python, berikut contoh penggunaan bahasa pemrograman python saat ini:

- a. *Google* menggunakan Python secara ekstensif dalam sistem pencarian webnya.
- b. yang populer *YouTube* sebagian besar ditulis dengan Python.
- c. Layanan *Dropbox* mengkode perangkat lunak klien desktop dan servernya terutama dengan Python.
- d. *Raspberry Pi* mempromosikan Python sebagai bahasa pendidikannya.
- e. yang tersebar luas *BitTorrent* memulai hidupnya sebagai program Python.
- f. *Google App Engine* menggunakan *Python* sebagai bahasa aplikasi.

² Puji Rahayu, "Pengaruh era digital terhadap perkembangan bahasa anak", *Jurnal metrouniv*, (2017): 47, <https://e-journal.metrouniv.ac.id/index.php/al-fathin/article/download/1423/1214/>

- g. *Maya*, sistem pemodelan dan animasi 3D terintegrasi yang kuat, menyediakan API skrip Python.
- h. *Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm,* dan *IBM* menggunakan Python untuk pengujian perangkat keras.
- i. *NASA, Los Alamos, Fermilab, JPL,* dan lainnya menggunakan Python untuk tugas pemrograman ilmiah.

Dampak negatif dari perkembangan teknologi adalah :

1. Penyalahgunaan komputer, misalnya penindasan dunia maya. penindasan dunia maya adalah ancaman, penghinaan, hingga komentar yang bertujuan untuk menindas orang lain. Selain penindasan dunia maya, dampak buruk penyalahgunaan komputer adalah peretasan, penipuan digital, juga penyebaran informasi yang salah.³
2. Kejahatan dunia maya (cyber crime) Seiring berkembangnya teknologi, berkembang pula kejahatan. Di dunia internet, kejahatan dikenal dengan nama cyber crime. Kejahatan dunia maya sangatlah beragam. Diantaranya, carding, hacking, cracking, phishing, dan spamming.
3. Melemahkan dan menurunkan sensitifitas Penurunan sensitifitas yang dimaksud disini adalah menurunnya tingkat simpati dan empati seseorang terhadap dunia nyata. Dengan jejaring sosial, seseorang cenderung melupakan dunia nyata dan tenggelam didalam dunia maya. Merenggangkan

³ Silmi Nurul Utami, "Dampak Perkembangan Komputer", Kompas.com, September 22, 2021, <https://www.kompas.com/skola/read/2021/09/22/113000369/dampak-perkembangankomputer?Page=all>.

dan mengabaikan sesuatu yang terjadi disekitarnya dan lebih memilih untuk memperhatikan sesuatu yang terjadi didunia maya.



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

BAB V

PENUTUP

A. Kesimpulan

Terciptanya bahasa pemrograman beriringan dengan sejarah mesin dan komputer. Awal mula bahasa pemrograman dimulai dari *Antikythera* yang berasal dari Yunani kuno. *Antikythera* adalah kalkulator yang menggunakan beberapa tuas dan konfigurasi untuk menjalankannya.

Bahasa pemrograman pertama adalah algoritma yang dibuat oleh Ada Lovelace pada tahun 1883. Ada Lovelace menciptakan algoritma ini untuk mesin Difference Engine Charles Babbage. Tujuan dari algoritma ini adalah untuk menghitung bilangan Bernoulli, yang diakui oleh beberapa sejarawan sebagai program komputer pertama yang diterbitkan di dunia.

Grace merancang sebuah compiler, yaitu perangkat yang menerjemahkan bahasa pemrograman yang dimengerti manusia, menjadi bahasa yang dimengerti komputer. Beliau mengaku bahwa dia adalah seorang yang malas, karena itu diciptakannya lah sebuah compiler agar pekerjaan pemrograman menjadi lebih mudah. Pada tahun 1957, Grace bersama dengan divisi yang dipimpinnya menelurkan kompuler A-0 generasi baru yang diberi nama Math-Matic. Math-Matic adalah compiler pertama yang menggunakan bahasa Inggris.

Bahasa pemrograman tingkat rendah adalah bahasa pemrograman yang menggunakan set instruksi khusus dari sebuah prosesor yang sudah ditentukan oleh pembuatnya. Bahasa pemrograman pertama berfungsi yang dirancang untuk mengkomunikasikan instruksi ke komputer ditulis pada awal 1950-an. Kode

Pendek John Mauchly, diusulkan pada tahun 1949, adalah salah satu bahasa tingkat tinggi pertama yang pernah dikembangkan untuk komputer elektronik. Tidak seperti kode mesin, pernyataan Kode Pendek mewakili ekspresi matematika dalam bentuk yang dapat dimengerti.

Namun, program harus diterjemahkan ke dalam kode mesin setiap kali dijalankan, membuat prosesnya jauh lebih lambat daripada menjalankan kode mesin yang setara. Kode mesin (bahasa mesin) adalah sistem instruksi untuk komputer tertentu, yang ditafsirkan langsung oleh prosesor komputer tersebut, didasarkan pada seperangkat perintah dasar yang dapat dimengerti oleh prosesor.

Bahasa pemrograman tingkat tinggi pertama adalah COBOL merupakan bahasa pemrograman yang didedikasi untuk keperluan bisnis, keuangan, dan sistem administrasi. Untuk keperluan sains, bahasa pemrograman yang diciptakan adalah Fortran yang dikembangkan oleh IBM. Bahasa pemrograman Fortran telah mengandung IF, DO, dan GOTO yang merupakan loncatan besar dalam perkembangan bahasa pemrograman. Pada tahun 1958, IBM merilis versi bahasa yang direvisi, bernama FORTRAN II.

Bahasa pemrograman Komputer sangat membantu, khususnya dalam menjalankan aktivitas, seperti belajar dan bekerja. Menyimpan dokumen atau data penting Selain memudahkan manusia, bahasa pemrograman komputer juga bermanfaat untuk menyimpan berbagai dokumen atau data penting, seperti hasil pekerjaan, tugas sekolah, dan foto.

dalam kehidupan sehari – hari, bahasa pemrograman komputer juga bermanfaat untuk melakukan perhitungan. Manfaat lain bahasa pemrograman

komputer bagi kehidupan masyarakat adalah pengolahan data. Contohnya program Microsoft Word untuk pengolahan data berupa teks, Microsoft Excel untuk pengolahan angka, Microsoft Power Point untuk pengolahan data dalam bentuk presentasi, dan masih banyak lagi. Bahasa pemrograman komputer juga bisa dimanfaatkan untuk menemukan suatu tempat. Pada saat ini bahasa pemrograman yang sangat bereperan penting dalam kamajuan teknologi adalah bahasa pemrograman Python,

B. Saran

saran saya dalam penelitian ini, untuk penelitian kedepanya agar lebih dikembangkan lagi dalam membahas bahasa pemrogrman tingkat rendah kerana masih banyak sekali kekurangan kerungan dalam sumber.

UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

DAFTAR PUSTAKA

Buku

- Ibnu Rasyid Munthe, dan Sudi Suryadi, 2018, *Bahasa Pemrograman Pascal*, Ponorogo: Uwais Inspirasi Indonesia.
- Horning, J. J. 1976, *History Of Programming Languages ii*. New York : ACM Sigplan.
- Horowitz, E. 1983, *The Evolution Of Programming Languages*. New York: Computer Science.
- Kasemin, Kasiyanto, 2015, *Agresi Perkembangan Teknologi Informasi*, Jakarta : PT Fajar Interpratama.
- Ledgord, H. 1986, *The World Of Programming Languages* . New York: Science Research Associates.
- Maclennan, Burce J. 1999, *Principles Of Programing Languages*, New York: Oxford University.
- Mitchell, John C. 2003, *Concepts in Programming Languages*, Cambridge University Press.
- Nuryanto, Hery, 2012, *Sejarah Perkembangan Teknologi Informasi*. Jakarta : PT Balai Pustaka.
- Sammet, J. e. 1996, *Programming Languages History And Fundamentalis* . New York: Prentice hall, Inc.
- Sugiyono, 2011, *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: Afabeta.
- Suparto. 2018. *bahasa pemrograman* . Jakarta: Departemen Pendidikan Nasional.
- Wexelblat, R. I. 1981. *History Of Programming Langues*. New York: Academic Press.

Artikel

Chamim, A. N. 2017, "Penggunaan Microcontroller Sebagai Pendektesi Posisi Dengan Menggunakan Sinyal GSM". *jurnal Informatika*. Vol. 4 ,No 1.

Dodi, Effend, 2014, "Kemudahan Pemrograman Mikrokontroler Arduino Pada Aplikasi Wahana Terbang" *Jurnal SETRUM* Vol. 3, No. 2 (<https://jurnal.untirta.ac.id/index.php/jis/article/view/507>).

Gunawan, I. 2019, *Pemrograman Assmbley Mikroprosesor* . Guepedia.

Kusuma, M. R, 2010. *Scribid*. Evolution of programming Languages : <https://id.scribd.com/doc/65375992/Evolusi-Bahasa-Pemrograman>.

Nerlove, Marc, 2013, "Programming Languages: A Short History for Economists", *The Journal of Economic and Social Measurement*.

Noah, N. 2017, *History and Evolution of Computer Programming Languages*. Diambil kembali dari Steemit: <https://steemit-com./steemstem/@noble-noah/history-and-evolution-of->.

Pratama, Muhammad Taufiq 2014 " Evolusi Bahasa Pemrograman ", *Jurnal Computech dan Bisnis*, Vol. 8 No. 1.

Sargi, R. R, 2018, "Pemrograman dan bahasa pemrograman", *Researchgate*. https://www.researchgate.net/publication/329885312_PEMROGRAMAN_DAN_BAHASA_PEMROGRAMAN?enrichId=rgreq.

Sharma, S, 2016, *Java Programing Language*. Diambil kembali dari Academia edu: https://www.academia.edu/10564468/java_Programming_Language.

Sholeh, Ahmad Timbul, 2013." Mengamankan Skrip Pada Bahasa Pemograman PHP Dengan Menggunakan Kriptografi Base64 " *Jurnal Algoritma Sekolah Tinggi Teknologi Garut*, Vol. 10 No. 1 (<http://jurnal.sttgarut.ac.id/>).

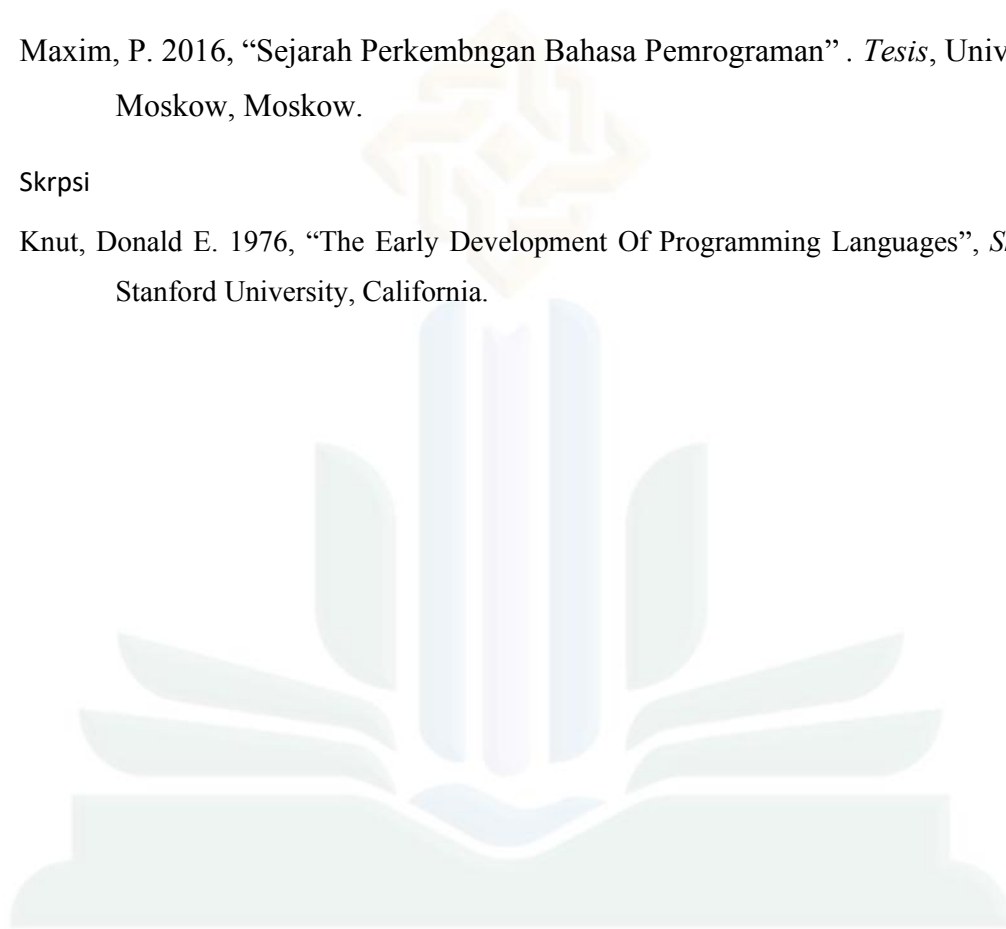
Vacic,Vladimir, 2012, "History of Programming Languages", *Jurnal University of California at Riverside*.

Tesis

Maxim, P. 2016, “Sejarah Perkembangan Bahasa Pemrograman” . *Tesis*, Universitas
Moskow, Moskow.

Skrpsi

Knut, Donald E. 1976, “The Early Development Of Programming Languages”, *Skripsi*,
Stanford University, California.



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

SURAT PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan di bawah ini:

Nama : Fitria Dewi Rochmawati

NIM : U20184025

Program Studi : Sejarah Peradaban Islam

Fakultas : Ushuluddin, Adab dan Humaniora

Institusi : Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember

Dengan ini menyatakan bahwa bagian atau keseluruhan isi Penulisan Skripsi ini tidak pernah diajukan untuk mendapatkan gelar akademis pada bidang studi dan atau universitas lain dan tidak pernah dipublikasikan atau ditulis oleh individu selain penyusun kecuali bila dituliskan dengan format kutipan dalam isi Penulisan Skripsi.

Apabila ditemukan bukti bahwa pernyataan saya tidak benar, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku di Universitas Islam Negeri Kiai Haji Achmad Siddiq Jember.

Jember, 14 Desember 2022

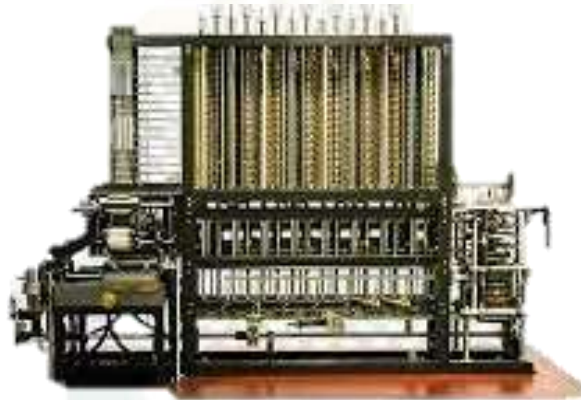
Saya yang menyatakan



Fitria Dewi Rochamawati

NIM U20184025

Lampiran – lampiran



Gambar mesin Difference Engine

Sumber: Ruang Guru, Minggu 31 Oktober 2021



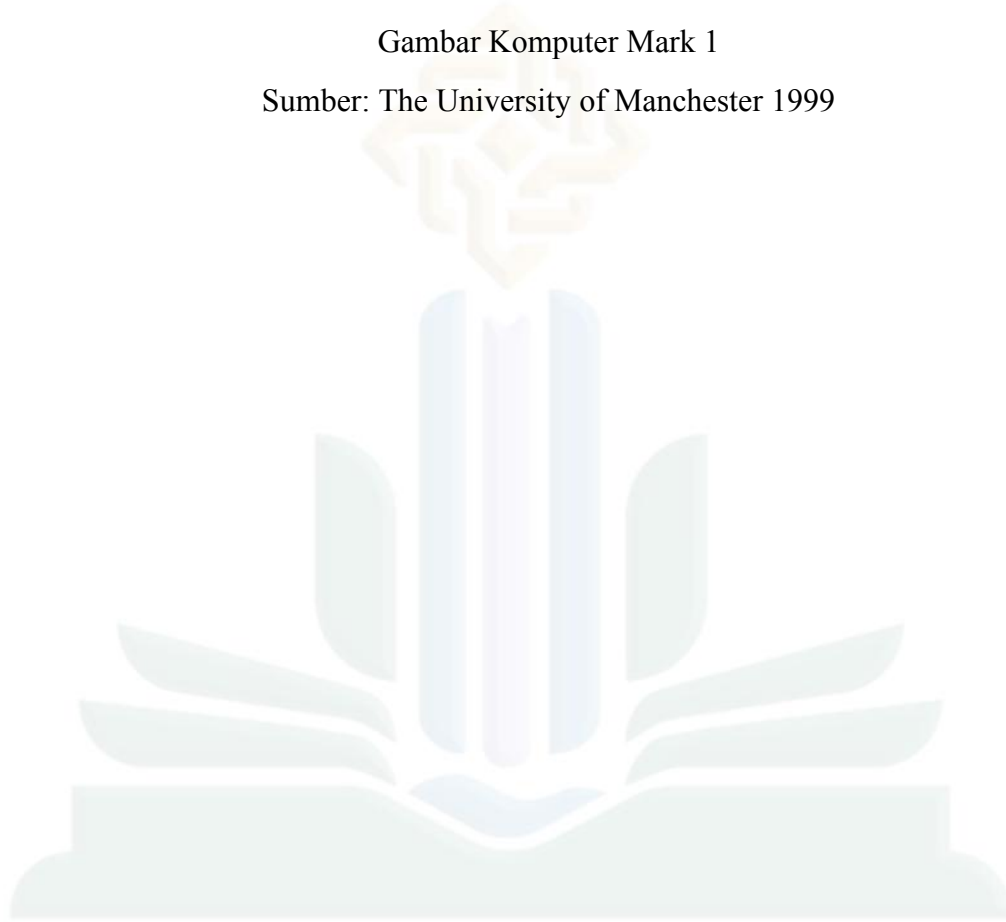
Gambar Mesin Komputer Z1 di buat ole Konrad Zuse

Sumber: WordPress, Kamis 29 Januari 2015



Gambar Komputer Mark 1

Sumber: The University of Manchester 1999



UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER

BIOGRAFI PENULIS



A. Identitas Diri

Nama : Fitria Dewi Rochmawati
Tempat/Tanggal Lahir : Sidoarjo, 9 Januari 2000
Jenis Kelamin : Perempuan
Alamat : Jl Raya Sukodono no 48, RT 03 RW 01
Fakultas : Ushuluddin, Adab dan Humaniora
Program Studi : Sejarah Peradaban Islam
NIM : U20184025

B. Riwayat Pendidikan

1. SDN Sukodono 1
2. SMP YPM 2 Sukodono
3. MAN 9 Jombang

C. Pengalaman Organisasi

1. IPPNU Sukodono

UIN

UNIVERSITAS ISLAM NEGERI

KH ACHMAD SIDDIQ
JEMBER