**STRUKTUR DATA DAN ALGORITMA**

**LAPORAN REVISI INDIVIDU PADA PROYEK AKHIR SEMESTER 2**

**Dosen Pengampu : Dr. Elly Matul Imah, M.Kom**

**Riskyana Dewi Intan Puspitasari, M.Kom**

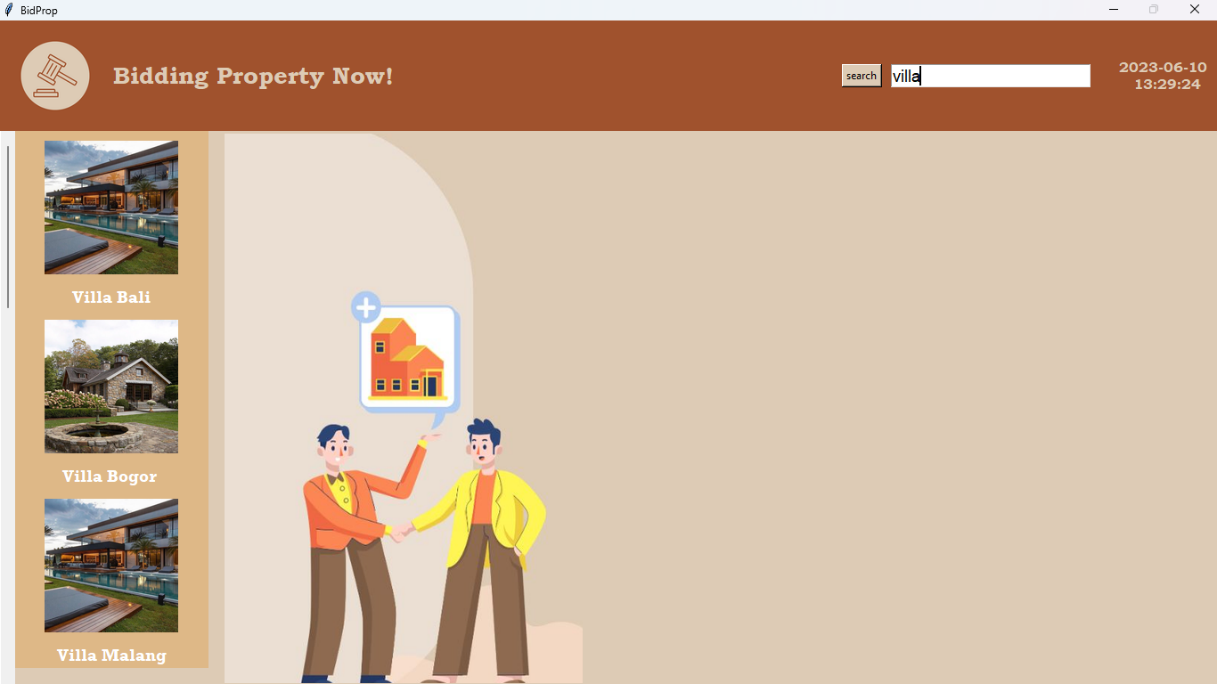
**Instansi : Universitas Negeri Surabaya**

**Nama : Joevita Salsabila Fitrianova**

**NIM : 22031554031**

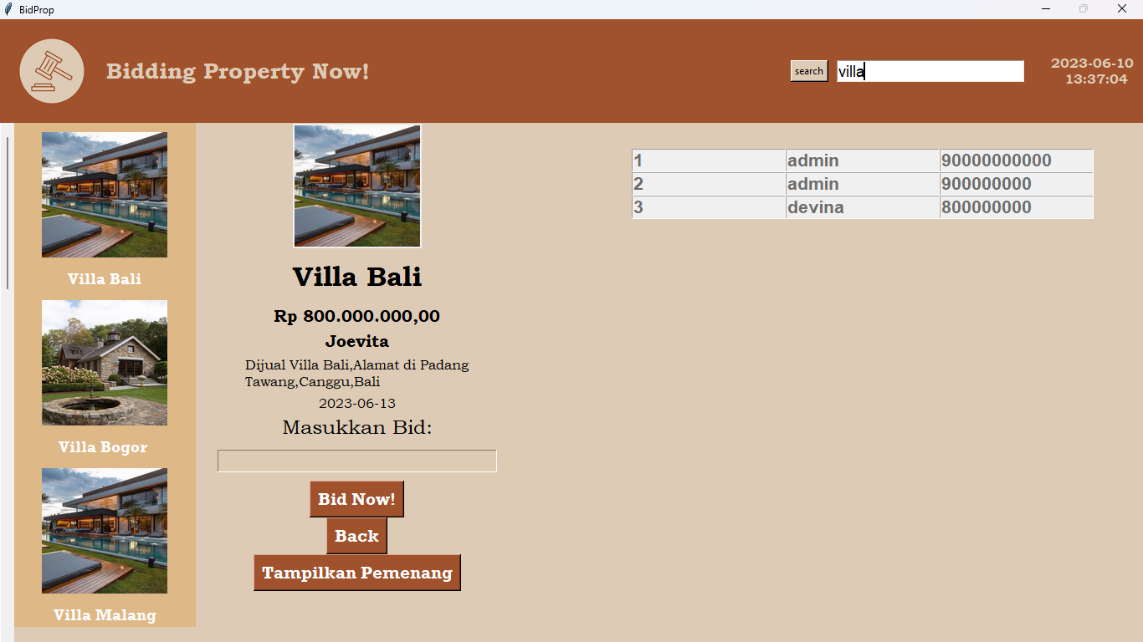
**Kelas : Sains Data 2022A**

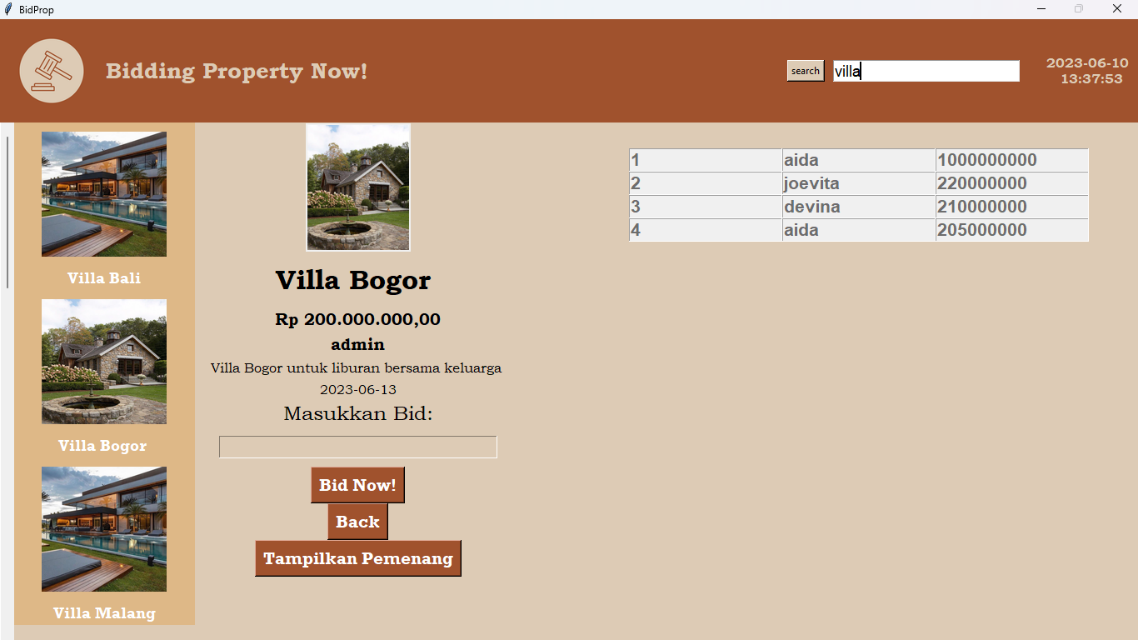
1. **Tugas Revisi**
2. Menambahkan algoritma pencarian binary Search pada proyek Lelang Property
3. Menambahkan combo box pada bagian pencarian yang berisi 2 pilihan, yakni linear search dan binary search.
4. Dapat mengurutkan item property berdasarkan harga secara ascending saat melakukan pencarian berdasarkan algoritma yang dipilih melalui combo box.
5. **Tampilan Sebelum Revisi**
6. Bagian pencarian sebelum revisi menggunakan algoritma linear searching



Pada bagian pencarian sebelum revisi menggunakan algoritma linear search. Jadi setiap user melakukan pencarian property, maka akan dilakukan pencarian properti menggunakan algoritma linear search.

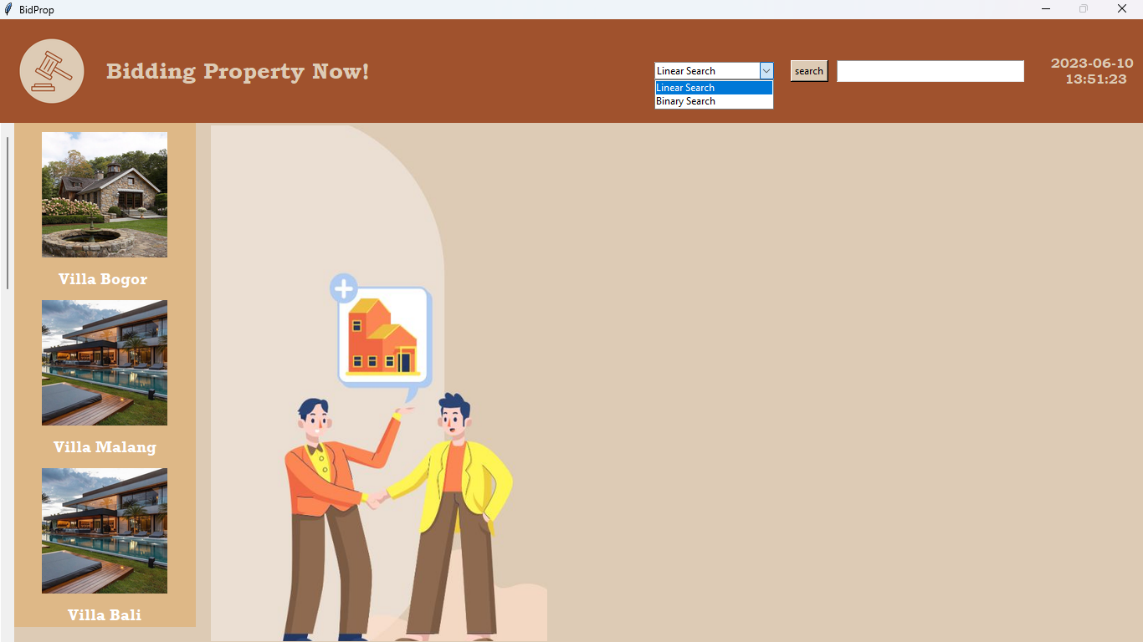
1. Hasil Pencarian menampilkan properti yang sesuai namun belum urut sesuai harga





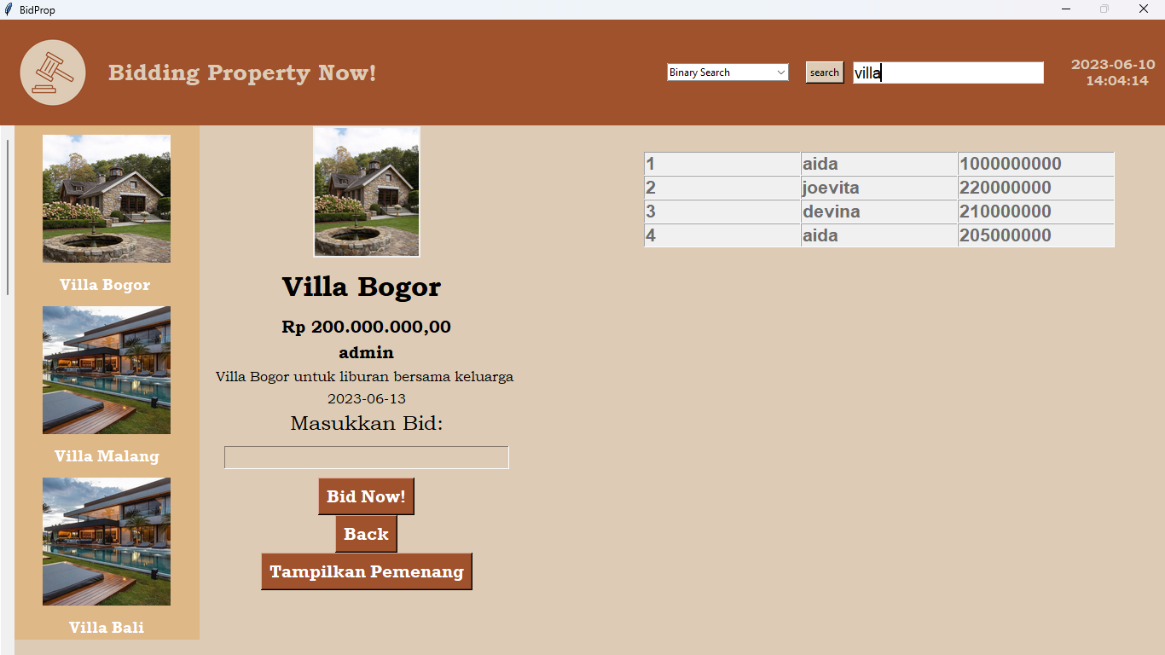
Berdasarkan pencarian yang dilakukan oleh user didapatkan hasil pencarian yang sesuai dengan keyword yang dicari, namun belum terurut berdasarkan harga property secara ascending.

1. **Tampilan Setelah Revisi**
2. Tampilan setelah ditambahkan algoritma binary search dan juga combo box



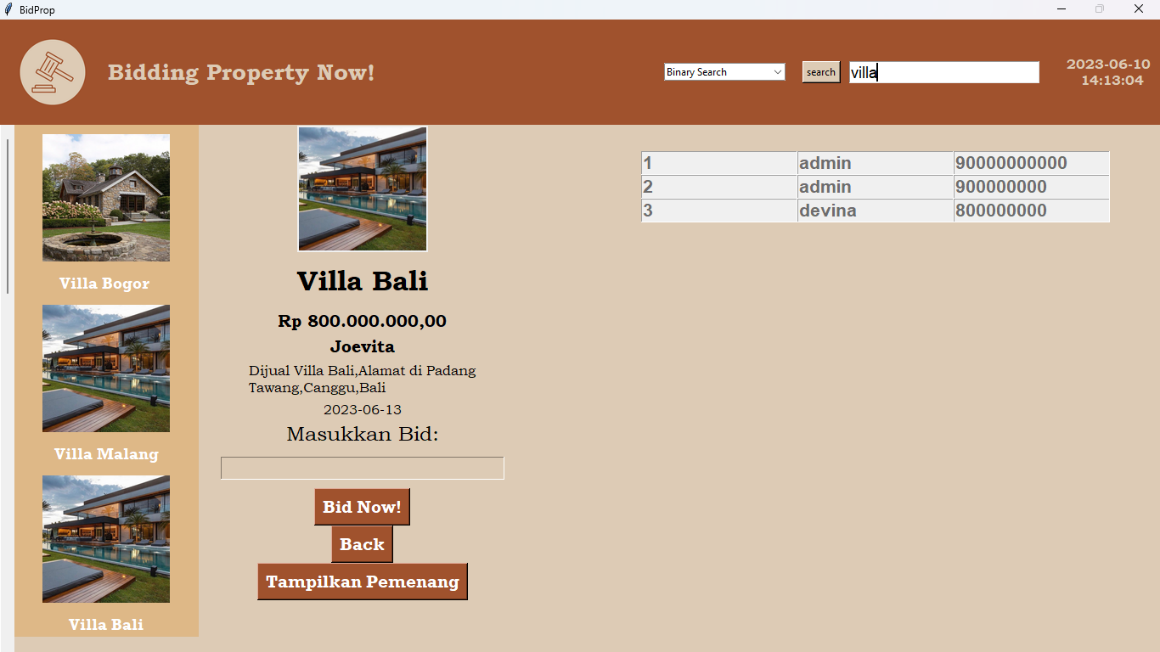
Pada bagian pencarian ditambahkan algoritma binary search yang digunakan untuk pencarian property. Selain itu, juga terdapat tampilan combo box yang memuat dua pilihan yakni linear search atau binary search yang nantinya akan digunakan untuk mencari property yang diinginkan user. User bebas memilih menggunakan algoritma linear search atau binary search dalam melakukan pencarian.

1. Tampilan saat dilakukan pencarian menggunakan Binary Search



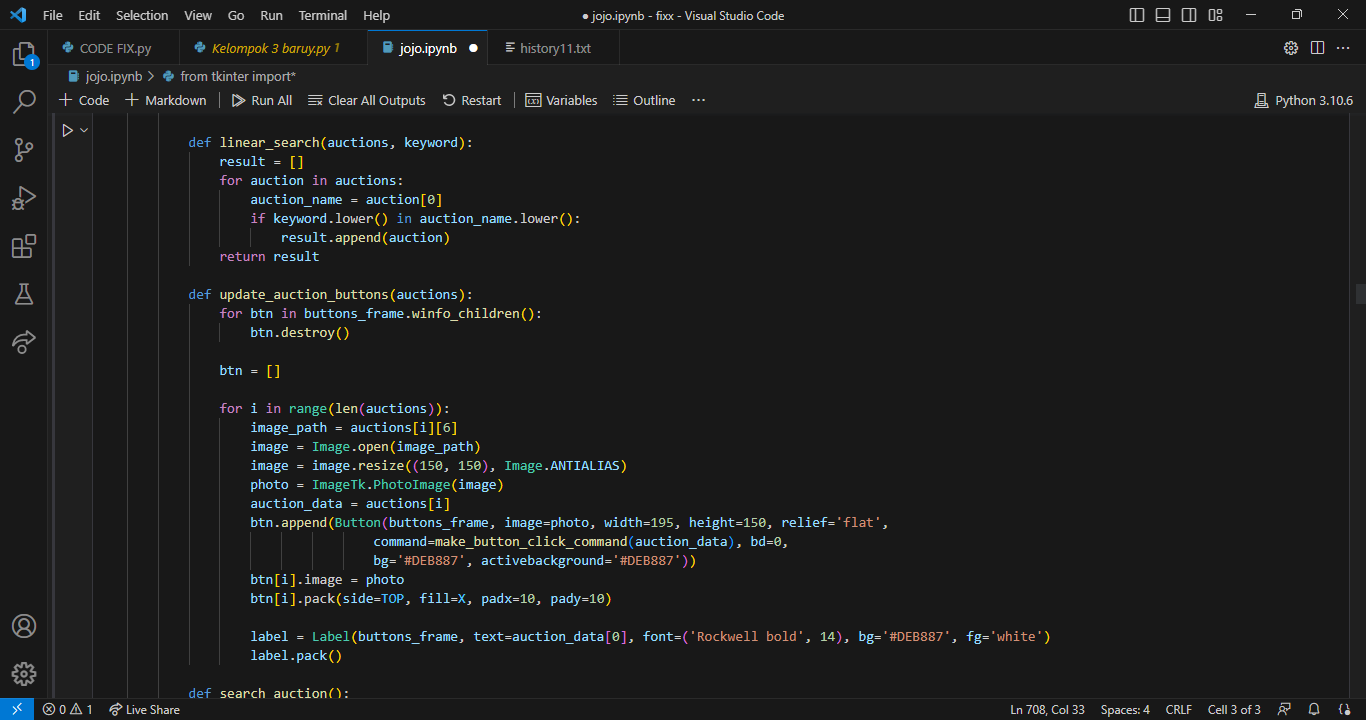
Pada bagian pencarian sebelumnya hanya menggunakan algoritma linear search, setelah dilakukan revisi, terdapat algoritma binary search untuk mencari properti yang diinginkan user dan hasil yang ditampilkan seperti pada gambar.

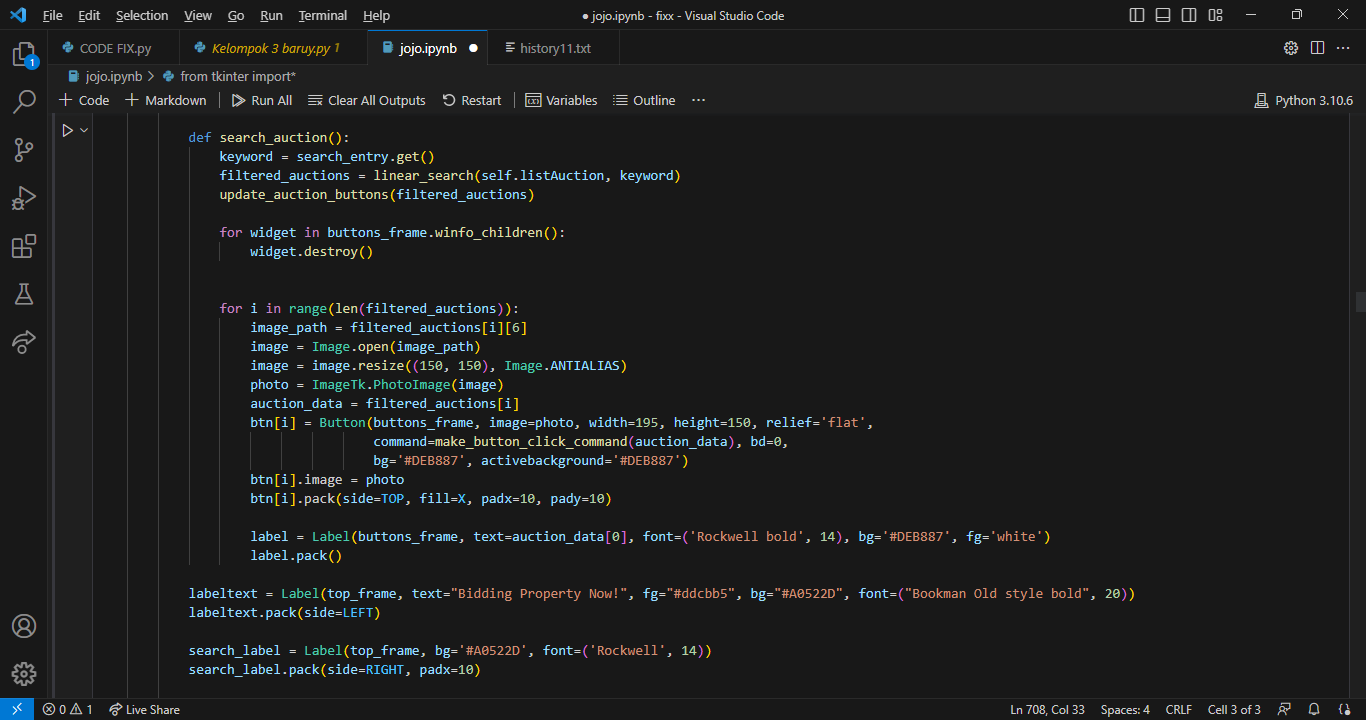
1. Hasil pencarian sudah terurut berdasarkan harga secara ascending



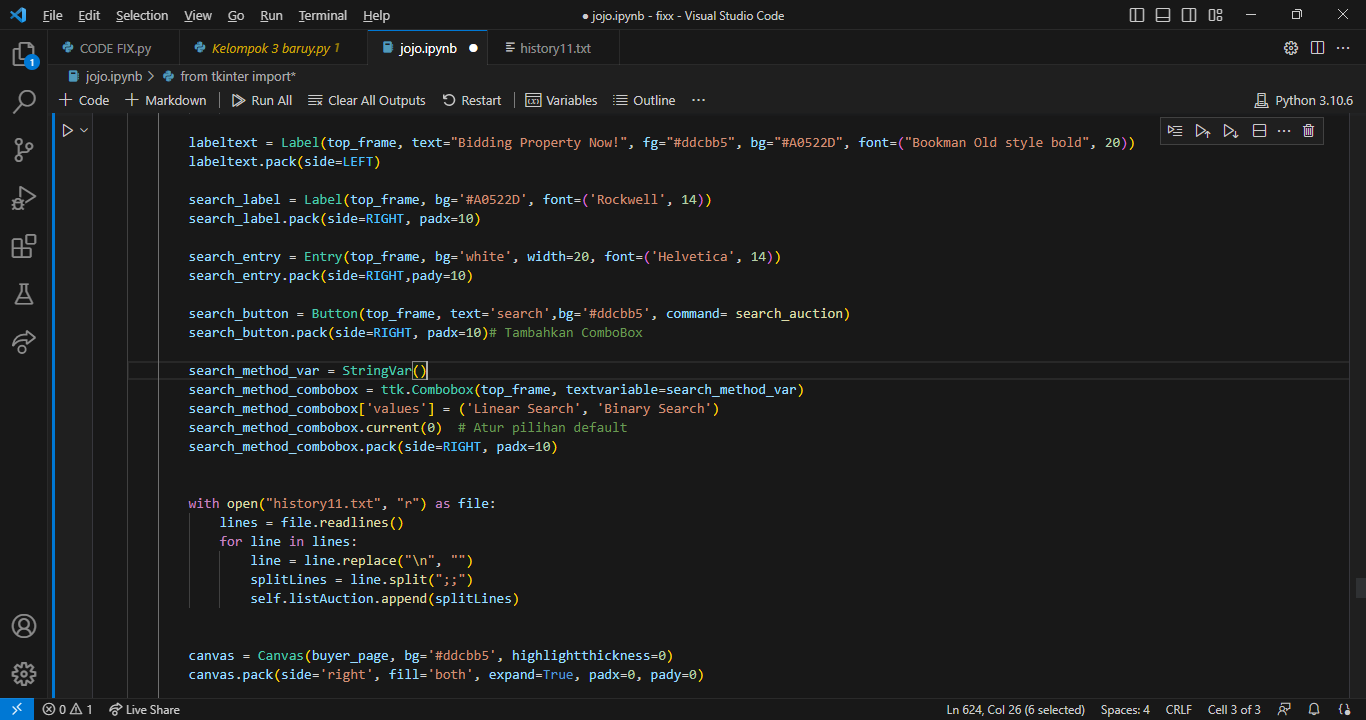
Hasil pencarian sebelumnya masih belum terurut secara ascending dan setelah dilakukan revisi, hasil pencarian berhasil diurutkan secara ascending berdasarkan harga properti. Hal tersebut dilakukan baik dengan menggunakan algoritma linear search atau binary search tetap terurut secara ascending.

1. **LISTING CODE**
2. Listing code asli sebelum revisi





1. Listing code setelah revisi

Menambahkan tampilan combo box pada pencarian

Menambahkan algoritma binary search

        def linear\_search(auctions, keyword, sort\_by\_price=False):

            result = []

            for auction in auctions:

                auction\_name = auction[0]

                if keyword.lower() in auction\_name.lower():

                    result.append(auction)

            if sort\_by\_price:

                result.sort(key=lambda x: x[1])  # Mengurutkan berdasarkan harga properti

            return result

        def binary\_search(auctions, keyword, sort\_by\_price=False):

            result = []

            left = 0

            right = len(auctions) - 1

            while left <= right:

                mid = (left + right) // 2

                auction\_name = auctions[mid][0]

                if keyword.lower() in auction\_name.lower():

                    # Cari ke kiri

                    i = mid

                    while i >= left and keyword.lower() in auctions[i][0].lower():

                        result.append(auctions[i])

                        i -= 1

                    # Cari ke kanan

                    i = mid + 1

                    while i <= right and keyword.lower() in auctions[i][0].lower():

                        result.append(auctions[i])

                        i += 1

                if keyword.lower() < auction\_name.lower():

                    right = mid - 1

                else:

                    left = mid + 1

            for auction in auctions:

                auction\_name = auction[0]

                if keyword.lower() in auction\_name.lower() and auction not in result:

                    result.append(auction)

            if sort\_by\_price:

                result.sort(key=lambda x: x[1])

            return result

        def search\_auction():

            keyword = search\_entry.get()

            search\_method = search\_method\_var.get()

            if search\_method == 'Linear Search':

                filtered\_auctions = linear\_search(self.listAuction, keyword, sort\_by\_price=True)

            else:

                # sorted\_auctions = sorted(self.listAuction, key=lambda x: x[1])

                filtered\_auctions = binary\_search(self.listAuction, keyword, sort\_by\_price=True)

            update\_auction\_buttons(filtered\_auctions)

            for widget in buttons\_frame.winfo\_children():

                widget.destroy()

            for i in range(len(filtered\_auctions)):

                image\_path = filtered\_auctions[i][6]

                image = Image.open(image\_path)

                image = image.resize((150, 150), Image.ANTIALIAS)

                photo = ImageTk.PhotoImage(image)

                auction\_data = filtered\_auctions[i]

                btn[i] = Button(buttons\_frame, image=photo, width=195, height=150, relief='flat',

                                command=make\_button\_click\_command(auction\_data), bd=0,

                                bg='#DEB887', activebackground='#DEB887')

                btn[i].image = photo

                btn[i].pack(side=TOP, fill=X, padx=10, pady=10)

                label = Label(buttons\_frame, text=auction\_data[0], font=('Rockwell bold', 14), bg='#DEB887', fg='white')

                label.pack()